ROBERT GORDON UNIVERSITY ABERDEEN

INFORMATICS INSTITUTE OF TECHNOLOGY

BSc (Hons) Artificial Intelligence and Data Science

Module: CM1603 Database Systems

Individual Coursework Report

Module Leader: Ms. Dileeka Alwis

RGU Student ID : 2409099

IIT Student ID : 20232838

Student Name : L.P.Vihanga Induwara warnapura

Acknowledgement

I would like to thank all those who supported me in creating and completing this report. I would also like to spread my heartfelt gratitude to our module leader, Ms. Dileeka Alwis for the constant guidance and support he gave me throughout this project. Without Ms. Dileeka Alwis and the other respective lecturers, I would not have been able to complete my coursework and report. Therefore, I would like to have a form of acknowledgment towards them in this report. I also want to express our gratitude to all the other IIT lecturers who helped us and guided us.

Finally,

I want to thank our parents and friends for helping us to complete this report.

Table of Contents Page No.

Section 01:

Assumptions

- Builders citadel have started on January 1, 2023.
- That there are only 10 employees in the company.
- That all employees are always involved in every project.
- If the money is paid in installments, no interest will be charged.
- One customer can sponsor several projects at once. But that there is only one customer for one project.
- That there are only supervisors and construction workers in the staff.
- Even though the supervisors are part of the staff, they have a separate id to identify them uniquely.
- That the staff will receive only monthly stipend. But supervisors are entitled to additional allowances.
- There are two payment methods for customers. They are installment payments and one time.
- That the customer is always an individual person or a company only.
- That the staff can work on several projects in the same week.
- That the time spent by each staff member on a separate project should be recorded every week.
- Every project done by builders citadel is a construction or renovation project.

Conceptual

- Entities:

The entities I have selected are as follows.

1. Customer
2. Project
3. Staff
4. Duty assignment
5. Payment

- Attributes:

The attributes I have selected for each entity.

1) Customer
   i. Customer id
   ii. Telephone
   iii. Name
      i) First name
      ii) last name
   iv. Type
      i) Customer
      ii) Company
   v. Addresses

2) Project

i. Project id
ii. Site location
iii. Actual competition date
iv. Start date
v. Estimated completion date
vi. Project type
vii. Specific progress

3) Staff

i. Staff id
ii. Salary
iii. Name
iv. Role
v. telephone number
vi. email address

4) Duty assignment

i. Staff id
ii. Project id
iii. Week id

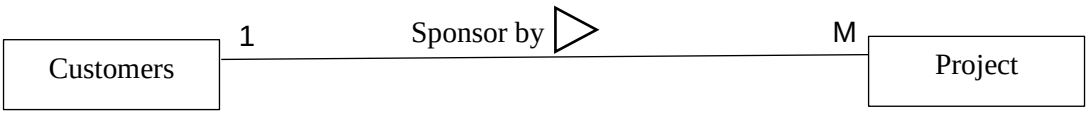
5) Payment

i. Payment id
ii. Type of payment
iii. Full amount
iv. Status

- Relationships:

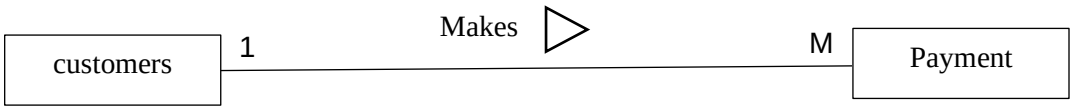
I have built relationships between entities as follows.

01).



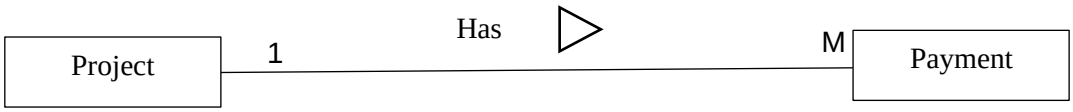
- One project belongs to one customer, and one customer may have multiple projects.

02).



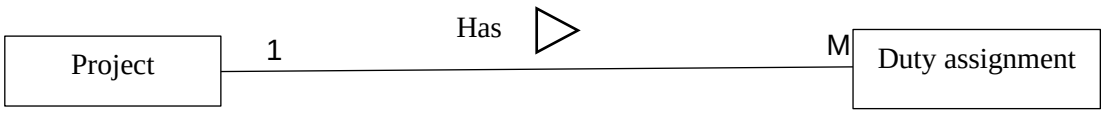
- One customer can make multiple payments, and each payment is associated with exactly one customer.

03).



- One project can involve multiple payments, and each payment is associated with exactly one project.

04).

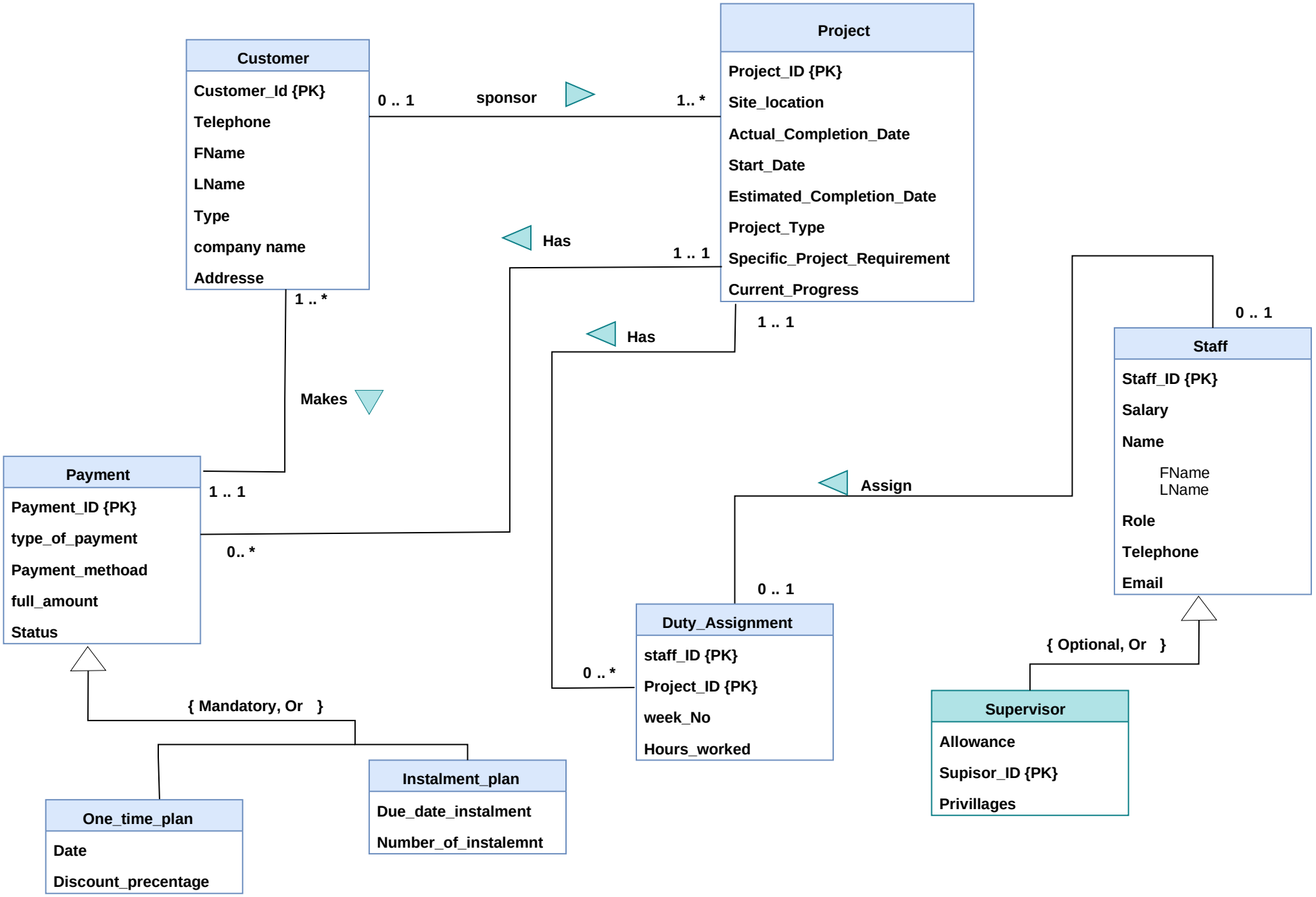


- One project may involve multiple duty assignments, and each duty assignment is associated with exactly one project.

05).

- One staff member can be assigned to multiple duties, and each duty assignment is associated with exactly one staff member.

Conceptual EER Diagram:


Customer
Customer_Id {PK}
Telephone
FName
LName
Type
company name
Addresse
0 .. 1
sponsor
1.. *
Project
Project_ID {PK}
Site_location
Actual_Completion_Date
Start_Date
Estimated_Completion_Date
Project_Type
Specific_Project_Requirement
Current_Progress
Has
1 .. 1
1 .. *
Makes
Has
1 .. 1
0 .. 1
Staff
Staff_ID {PK}
Salary
Name
FName
LName
Role
Telephone
Email
Payment
Payment_ID {PK}
type_of_payment
Payment_methoad
full_amount
Status
1 .. 1
0.. *
Assign
0 .. 1
Duty_Assignment
staff_ID {PK}
Project_ID {PK}
week_No
Hours_worked
0 .. *
{ Optional, Or }
Supervisor
Allowance
Supisor_ID {PK}
Privillages
{ Mandatory, Or }
One_time_plan
Date
Discount_precentage
Instalment_plan
Due_date_instalment
Number_of_instalemnt

Section 01 – logical:

Entities:

The entities I have selected are as follows.

1. Customer
2. Project
3. Staff
4. Duty assignment
5. Supervisor
6. Payment
7. One-time payment
8. Installment payment

Attributes:

1) Customer

i. Customer_ID {Primary key}
ii. FName
iii. LName
iv. Company_name
v. Telephone
vi. Type
vii. Addresse

2) Project

i. Project_id {Primary key}
ii. Customer_id {foreign key from customer table}
iii. Site_location
iv. week_no
v. actual_completion_date
vi. start_date
vii. estimated_completion_,
viii. project_type
ix. specific_project_requirement
x. current_progress

3) staff

i. staff_id {Primary key}
ii. salary
iii. FName
iv. LName
v. Role
vi. Telephone
vii. Email

4) Supervisor

i. staff_id {foreign key from staff table}
ii. allowance
iii. Supervisor_id {Primary key}
iv. Privileges

5) duty assignment

i. duty_assignment_id {Primary key}
ii. project_id {foreign key from project table}
iii. staff_id {foreign key from staff table}
iv. current_week_no
v. hours_worked

6) payment_instalement_plan

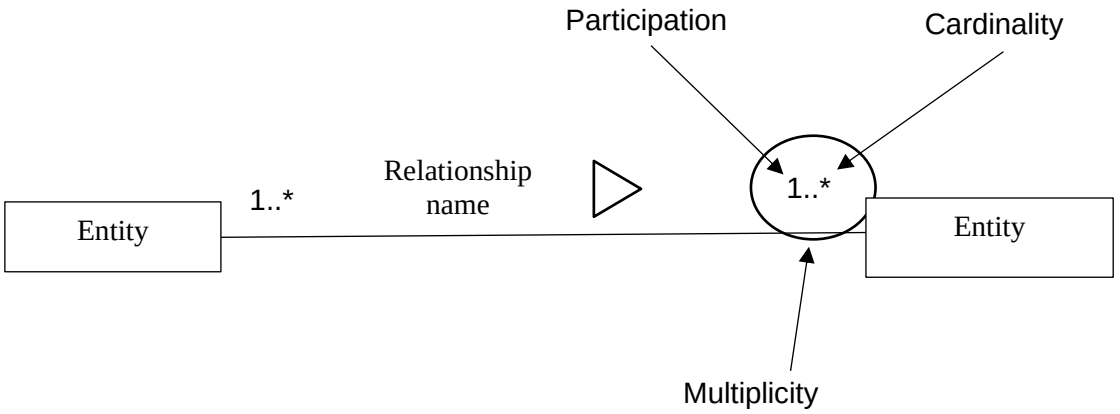
i. instalement_id {Primary key}
ii. project_id {foreign key from project table}
iii. customer_id {foreign key from customer table}
iv. payment_methoad
v. type_of_payment
vi. full_amount
vii. status_
viii. number_of_instalemnt

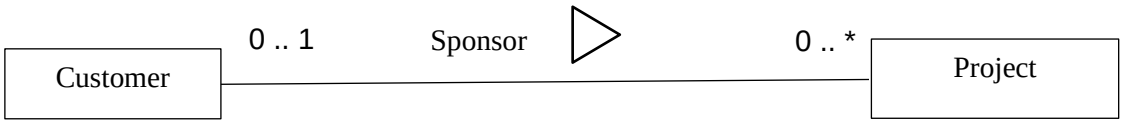
7) payment_one_time_plan

i. one_time_id {Primary key}
ii. project_id {foreign key from project table}
iii. customer_id {foreign key from customer table}
iv. payment_methoad
v. type_of_payment
vi. full_amount
vii. status_
viii. paid_Date
ix. discount_precentage

- Relationships

Ex:



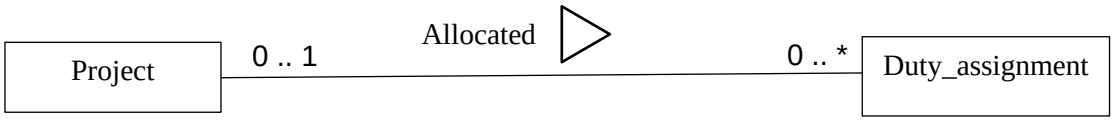


- Participation:

Each customer must be associated with at least one project (mandatory participation).
Each project may or may not be associated with a customer (optional participation).

- Cardinality:

One customer can sponsor many projects.
One project can be sponsored by only one customer.

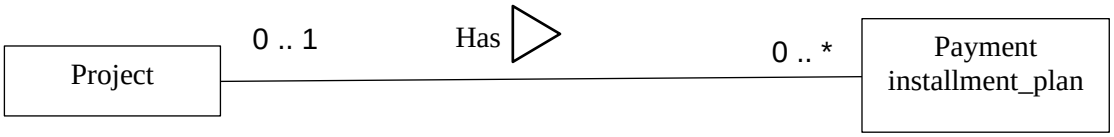


- Participation:

Each project must have at least one duty assignment (mandatory participation).
Each duty assignment must be associated with exactly one project (mandatory participation).

- Cardinality:

One project can have many duty assignments.
One duty assignment belongs to only one project.



- Participation:

Each project may or may not have a payment installment plan associated with it (optional participation).
Each payment installment plan must be associated with exactly one project (mandatory participation).

- Cardinality:

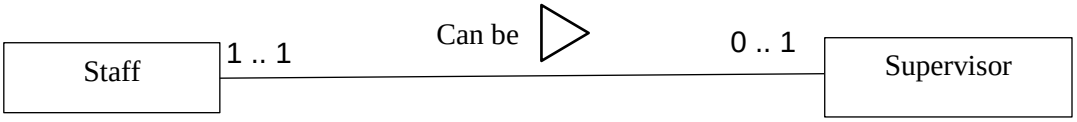
One project can have at most one payment installment plan.
One payment installment plan belongs to only one project.

- Participation:

Each project may or may not have a payment one-time plan associated with it (optional participation).
Each payment one-time plan must be associated with exactly one project (mandatory participation).

- Cardinality:

One project can have at most one payment one-time plan.
One payment one-time plan belongs to only one project.



- Participation:

Each staff member may or may not have a supervisor (optional participation).
Each supervisor must supervise at least one staff member (mandatory participation).

- Cardinality:

One supervisor can supervise many staff members.
One staff member may have at most one supervisor, but they might have none.

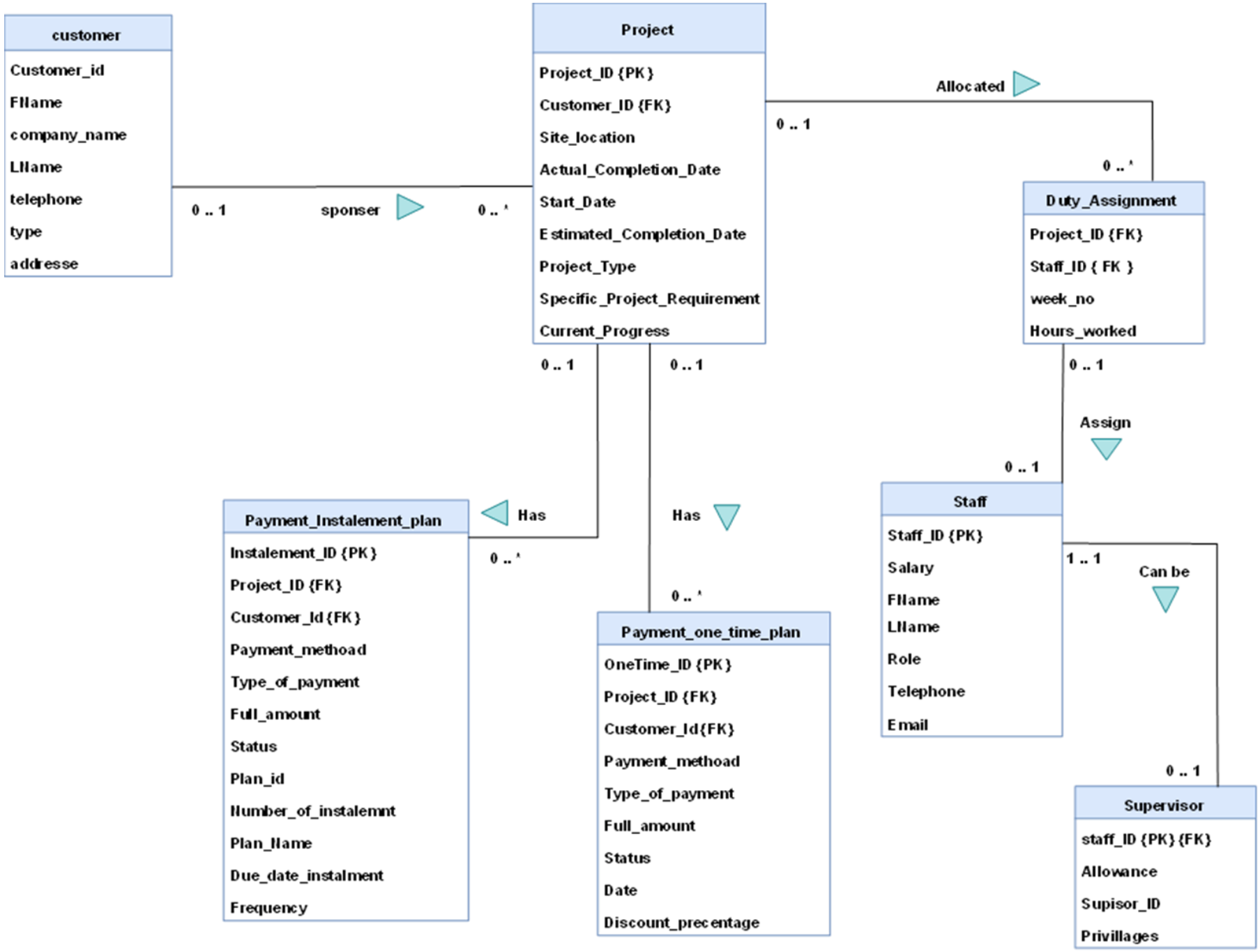
- Participation:

Each duty assignment must be assigned to at least one staff member (mandatory participation).
Each staff member may or may not have duty assignments (optional participation).

- Cardinality:

One duty assignment can be assigned to one or more staff members.
One staff member can be assigned to many duty assignments.

EER Diagram – logical:


customer
Customer_id
FIName
company_name
LIName
telephone
type
addresse
0 .. 1
sponser
0 .. *
Project
Project_ID {PK}
Customer_ID {FK}
Site_location
Actual_Completion_Date
Start_Date
Estimated_Completion_Date
Project_Type
Specific_Project_Requirement
Current_Progress
Allocated
0 .. 1
0 .. *
Duty_Assignment
Project_ID {FK}
Staff_ID { FK }
week_no
Hours_worked
0 .. 1
0 .. 1
Assign
0 .. 1
Staff
Staff_ID {PK}
Salary
FIName
LIName
Role
Telephone
Email
1 .. 1
Can be
0 .. 1
Supervisor
staff_ID {PK} {FK}
Allowance
Supisor_ID
Privillages
Payment_Instalement_plan
Instalement_ID {PK}
Project_ID {FK}
Customer_Id {FK}
Payment_methoad
Type_of_payment
Full_amount
Status
Plan_id
Number_of_instalemnt
Plan_Name
Due_date_instalment
Frequency
Has
0 .. *
Has
0 .. *
Payment_one_time_plan
OneTime_ID {PK}
Project_ID {FK}
Customer_Id{FK}
Payment_methoad
Type_of_payment
Full_amount
Status
Date
Discount_precentage

Section 3:

Creation

- Create Database

```
CREATE DATABASE Builders_Citadel;
```

CREATE DATABASE: This is a SQL statement used to create a new database.

Builders_Citadel: This is the name given to the new database.
It appears to be named "Builders_Citadel" in this case.

- Create customer table

```
CREATE TABLE customer(
    customer_no     char(10),
    FName   varchar(50),
    LName   varchar(50),
    company_name    varchar(50),
    telephone       char(10) NOT NULL,
    type ENUM('person','company'),
    addresse    varchar(150),
    PRIMARY KEY(customer_no)
);
```

- CREATE TABLE customer: This line initiates the creation of new table named customer.
- (...): The parentheses close the list of columns and their definitions within the table.
- customer_id char(10*):* This column is intended to store the customer's ID, represented as a string of up to 10 characters.
- FName varchar (50): This column stores the customer's first name as a variable-

length string of up to 50 characters.

- LName varchar (50): This column stores the customer's last name as a variable-length string of up to 50 characters.
- company_name varchar(50): This column is for storing the name of the customer's company as a variable-length string of up to 50 characters.
- telephone char (10) NOT NULL: This column stores the customer's telephone number as a string of up to 10 characters. The NOT NULL constraint ensures that this field must have a value for every record.
- type ENUM('person','company'): This column defines the type of customer, either 'person' or 'company'. It uses the ENUM data type, which restricts the values that can be inserted into this column to only those specified ('person' or 'company').
- addresse varchar (150): This column is intended to store the customer's address as a variable-length string of up to 150 characters.
- PRIMARY KEY (customer_no): This line specifies the primary key for the table. However, there seems to be a typo (customer_no instead of customer_id). It should likely be PRIMARY KEY (customer_id) if customer_id is intended to be the primary key.

- Create project table

```sql
CREATE TABLE project (
    project_id CHAR(5),
    customer_id CHAR(5),
    site_location VARCHAR(50) NOT NULL,
    week_no int(9),
    actual_completion_date DATE,
    start_date DATE NOT NULL,
    estimated_completion_date DATE,
    project_type ENUM('Construction', 'Renovation'),
    specific_project_requirement VARCHAR(250),
    current_progress  ENUM('Planing', 'ongoing','completed'),
    PRIMARY KEY (project_id),
    FOREIGN KEY (customer_id) REFERENCES customer(customer_id)
);
```

** The 'planing' section was later removed*

- CREATE TABLE project: This line initiates the creation of a new table named project.
- (...): The parentheses enclose the list of columns and their definitions within the table.

- project_id CHAR(5): This column stores the project's ID, represented as a string of up to 5 characters.
- customer_id CHAR(5): This column stores the ID of the customer associated with the project, represented as a string of up to 5 characters. It is intended to be a foreign key referencing the customer table's customer_id column.
- site_location VARCHAR(50) NOT NULL: This column stores the location of the project site as a variable-length string of up to 50 characters. The NOT NULL constraint ensures that this field must have a value for every record.
- week_no INT(9): This column stores the week number of the project, represented as an integer.
- actual_completion_date DATE: This column stores the actual completion date of the project.
- start_date DATE NOT NULL: This column stores the start date of the project. The NOT NULL constraint ensures that this field must have a value for every record.
- estimated_completion_date DATE: This column stores the estimated completion date of the project.
- project_type ENUM('Construction', 'Renovation'): This column defines the type of project, either 'Construction' or 'Renovation'. It uses the ENUM data type to restrict the values that can be inserted into this column.
- specific_project_requirement VARCHAR(250): This column stores specific requirements or details about the project as a variable-length string of up to 250 characters.
- current_progress ENUM('Planning', 'ongoing', 'completed'): This column indicates the current progress status of the project, which can be 'Planning', 'ongoing', or 'completed'. It uses the ENUM data type to restrict the values that can be inserted into this column.
- PRIMARY KEY (project_id): This line specifies the primary key for the table, which is the project_id column.
- FOREIGN KEY (customer_id) REFERENCES customer(customer_id): This line establishes a foreign key constraint on the customer_id column, referencing the customer table's customer_id column. This ensures referential integrity between the project and customer tables, meaning that every project record must correspond to a valid customer record.

- Create staff table

```sql
CREATE TABLE Staff(
    staff_id char(5),
    salary int(8),
    FName varchar(50) NOT NULL,
    LName varchar(50) NOT NULL,
    role ENUM('Construction Worker','Supervisor'),
    telephone char(10),
    email varchar(50),
    PRIMARY KEY(staff_id)
);
```

- CREATE TABLE Staff: This line initiates the creation of a new table named Staff.
- (...): The parentheses enclose the list of columns and their definitions within the table.
- staff_id CHAR(5): This column stores the staff member's ID, represented as a string of up to 5 characters.
- salary INT(8): This column stores the staff member's salary as an integer, with a maximum display width of 8 digits.
- FName VARCHAR(50) NOT NULL: This column stores the staff member's first name as a variable-length string of up to 50 characters. The NOT NULL constraint ensures that this field must have a value for every record.
- LName VARCHAR(50) NOT NULL: This column stores the staff member's last name as a variable-length string of up to 50 characters. The NOT NULL constraint ensures that this field must have a value for every record.
- role ENUM('Construction Worker', 'Supervisor'): This column defines the role of the staff member, which can be either 'Construction Worker' or 'Supervisor'. It uses the ENUM data type to restrict the values that can be inserted into this column.
- telephone CHAR(10): This column stores the staff member's telephone number as a string of up to 10 characters.

- email VARCHAR(50): This column stores the staff member's email address as a variable-length string of up to 50 characters.
- PRIMARY KEY (staff_id): This line specifies the primary key for the table, which is the staff_id column. This ensures that each staff member has a unique identifier within the table.

- Create supervisor table

```
CREATE TABLE supervisor(
    staff_id char(5),
    allowance int(8),
    supisor_id char(5),
    privillages varchar(200),

    PRIMARY KEY(supisor_id),
    FOREIGN KEY(staff_id) REFERENCES staff(staff_id)
);

```

- CREATE TABLE supervisor: This line initiates the creation of a new table named supervisor.
- (...): The parentheses enclose the list of columns and their definitions within the table.
- staff_id CHAR(5): This column stores the ID of the staff member who is a supervisor, represented as a string of up to 5 characters. It serves as a foreign key referencing the staff_id column in the staff table.
- allowance INT(8): This column stores the allowance amount for the supervisor, represented as an integer with a maximum display width of 8 digits.
- supisor_id CHAR(5): This column stores the ID of the supervisor, represented as a string of up to 5 characters. It is the primary key of the supervisor table.
- privillages VARCHAR(200): This column stores the privileges or rights granted to the

supervisor as a variable-length string of up to 200 characters.

- PRIMARY KEY(supisor_id): This line specifies the primary key for the table, which is the supisor_id column. This ensures that each supervisor has a unique identifier within the table.
- FOREIGN KEY(staff_id) REFERENCES staff(staff_id): This line establishes a foreign key constraint on the staff_id column, referencing the staff table's staff_id column. It ensures referential integrity between the supervisor and staff tables, meaning that every supervisor record must correspond to a valid staff record.

- Create duty assignment table

```
CREATE TABLE duty_assignment (
    duty_assignment_no CHAR(5),
    project_id CHAR(5),
    staff_id CHAR(5),
    current_week_no INT(9),
    hours_worked INT NOT NULL,

    PRIMARY KEY(project_id, staff_id),
    FOREIGN KEY(project_id) REFERENCES project(project_id),
    FOREIGN KEY(staff_id) REFERENCES staff(staff_id)
);
```

- `CREATE TABLE duty_assignment`: The creation account begins with `duty_assignment` being a dedicated table that holds the primary data.
- `( ... )`: Within the brackets the list of columns and their view is given.
- `duty_assignment_id CHAR(5)`: This variable contains the duty assignment ID value represented by a string of shapes up to 5 characters in length. It is the primary value of the `duty_assignment table` because it has related codes in factors and existing facilities.

- `project_id CHAR(5)`: This field is where the project id of the workAssignment is stored as a character string of upto 5. It is a way for it to act as a foreign key and reference the `project_id` column in the `project` table.
- `staff_id CHAR(5)`: Duty ID for the staff member is stored in this field which can be either a string value with a maximum length of 5 characters or any other valid data type. It will act as a foreign key that will refer to the `staff_id` column at the `staff` table.
- `current_week_no INT(9)`: This field refers to the current week number of conduction work/task, represented by an integer.
- `hours_worked INT NOT NULL`: This field keeps for the quantity of the accounted hours for for the certain assignment, which is measured by integer. This `NOT NULL` restraint guarantees that the same field can`t be empty for every record.
- `PRIMARY KEY(duty_assignment_id)`: Here the table gets the primary key determined by the unique identity column `duty_assignment_id`. Through this act, the table has the distinction of being equipped with a special tag that identifies each one-of-a-kind duty.
- FOREIGN KEY(project_id) REFERENCES project(project_id): This condition defines a foreign key constraint on the project_id column, referring to the project_id column from the project table. This guarantees the referential integrity across the duty_assignment and project tables, wherein each respective duty assignment record must correspond to a validity.
- FOREIGN KEY(staff_id) REFERENCES staff(staff_id): This is where we mention a FOREIGN KEY constraint on the staff_id column that will be consistent with the staff table's staff_id column. It guarantees the referential integrity between the duty_assignment and staff tables such that each entry in the former table is associated with a valid employee entry in the latter table.

- Create Payment installment plan table

```sql
CREATE TABLE payment_instalement_plan(
    instalement_id char(5),
    project_id char(5),
    customer_id char(5),
    payment_methoad ENUM('Cash', 'Credit Card', 'Bank Transfer', 'Check'),
    type_of_payment varchar(12) DEFAULT 'instalement',
    full_amount int(10),
    status_ ENUM('Pending','Finshed'),
    number_of_instalemnt int(3),
    Plan_Name varchar(20),
    due_date_instalment date,
    PRIMARY KEY(instalement_id),
    FOREIGN KEY(project_id) REFERENCES project(project_id),
    FOREIGN KEY(customer_id) REFERENCES customer(customer_id)
);
```

- `CREATE TABLE payment_instalment_plan`: This line initiates the creation of a new table named `payment_instalment_plan`.
- `( ... )`: The parentheses enclose the list of columns and their definitions within the table.
- `instalment_id CHAR(5)`: This column stores the ID of the instalment, represented as a string of up to 5 characters. It is the primary key of the `payment_instalment_plan` table.
- `project_id CHAR(5)`: This column stores the ID of the project associated with the instalment, represented as a string of up to 5 characters. It serves as a foreign key referencing the `project_id` column in the `project` table.
- `customer_id CHAR(5)`: This column stores the ID of the customer associated with the instalment, represented as a string of up to 5 characters. It serves as a foreign key referencing the `customer_id` column in the `customer` table.
- `payment_method ENUM('Cash', 'Credit Card', 'Bank Transfer', 'Check')`: This column represents the payment method for the instalment and is defined as an

ENUM data type, restricting the values that can be inserted to 'Cash', 'Credit Card', 'Bank Transfer', or 'Check'.

- `type_of_payment VARCHAR(12) DEFAULT 'instalment'`: This column represents the type of payment, which is set to 'instalment' by default. It's a variable-length string of up to 12 characters.
- `full_amount INT(10)`: This column stores the full amount of the instalment, represented as an integer with a maximum display width of 10 digits.
- `status_ ENUM('Pending', 'Finished')`: This column represents the status of the instalment, which can be either 'Pending' or 'Finished'. It's defined as an ENUM data type.
- `number_of_instalment INT(3)`: This column stores the number of instalments, represented as an integer with a maximum display width of 3 digits.
- `PRIMARY KEY(instalement_id)`: This line specifies the primary key for the table, which is the `instalment_id` column. This ensures that each instalment has a unique identifier within the table.
- `FOREIGN KEY(project_id) REFERENCES project(project_id)`: This line establishes a foreign key constraint on the `project_id` column, referencing the `project` table's `project_id` column. It ensures referential integrity between the `payment_instalment_plan` and `project` tables, meaning that every record in `payment_instalment_plan` must correspond to a valid record in `project`.

- `FOREIGN KEY(customer_id) REFERENCES customer(customer_id)`: This line establishes a foreign key constraint on the `customer_id` column, referencing the `customer` table's `customer_id` column. It ensures referential integrity between the `payment_instalment_plan` and `customer` tables, meaning that every record in `payment_instalment_plan` must correspond to a valid record in `customer`.

- Create one time payment table

```sql
CREATE TABLE payment_one_time_plan(
    one_time_id char(5),
    project_id char(5),
    customer_id char(5),
    payment_methoad ENUM('Cash', 'Credit Card', 'Bank Transfer', 'Check'),
    type_of_payment varchar(12) DEFAULT 'one time',
    full_amount int(10),
    status_ ENUM('Pending','Countinue','Finshed'),
    paid_Date date,
    discount_precentage char(4),
    PRIMARY KEY(one_time_id),
    FOREIGN KEY(project_id) REFERENCES project(project_id),
    FOREIGN KEY(customer_id) REFERENCES customer(customer_id)
);

```

- CREATE TABLE payment_one_time_plan: This line initiates the creation of a new table named payment_one_time_plan.
- (...): The parentheses enclose the list of columns and their definitions within the table.
- one_time_id CHAR(5): This column stores the ID of the one-time payment plan, represented as a string of up to 5 characters. It is the primary key of the payment_one_time_plan table.
- project_id CHAR(5): This column stores the ID of the project associated with the one-time payment plan, represented as a string of up to 5 characters. It serves as a foreign key referencing the project_id column in the project table.
- customer_id CHAR(5): This column stores the ID of the customer associated with the one-time payment plan, represented as a string of up to 5 characters. It serves as a foreign key referencing the customer_id column in the customer table.
- payment_method ENUM('Cash', 'Credit Card', 'Bank Transfer', 'Check'): This

column represents the payment method for the one-time payment plan and is defined as an ENUM data type, restricting the values that can be inserted to 'Cash', 'Credit Card', 'Bank Transfer', or 'Check'.

- type_of_payment VARCHAR(12) DEFAULT 'one time': This column represents the type of payment, which is set to 'one time' by default. It's a variable-length string of up to 12 characters.
- full_amount INT(10): This column stores the full amount of the one-time payment plan, represented as an integer with a maximum display width of 10 digits.
- status_ ENUM('Pending', 'Continue', 'Finished'): This column represents the status of the one-time payment plan, which can be 'Pending', 'Continue', or 'Finished'. It's defined as an ENUM data type.
- paid_date DATE: This column stores the date when the payment was made for the one-time payment plan.
- discount_percentage CHAR(4): This column stores the discount percentage applied to the one-time payment plan, represented as a string of up to 4 characters.
- PRIMARY KEY(one_time_id): This line specifies the primary key for the table, which is the one_time_id column. This ensures that each one-time payment plan has a unique identifier within the table.
- FOREIGN KEY(project_id) REFERENCES project(project_id): This line establishes a foreign key constraint on the project_id column, referencing the project table's project_id column. It ensures referential integrity between the payment_one_time_plan and project tables, meaning that every record in payment_one_time_plan must correspond to a valid record in project.
- FOREIGN KEY(customer_id) REFERENCES customer(customer_id): This line establishes a foreign key constraint on the customer_id column, referencing the customer table's customer_id column. It ensures referential integrity between the payment_one_time_plan and customer tables, meaning that every record in payment_one_time_plan must correspond to a valid record in customer.

Population of data

- Data insert into customer table

```
INSERT INTO customeR(customer_id, FName, LName, company_name, Telephone, Type, Addresse)
VALUES
    ('C001', 'Nimal', 'Silva', NULL, '0771234567', 'Person', 'No: 10, Main Street, Colombo'),
    ('C002', 'Kamal', 'Perera', NULL, '0712345678', 'Person', 'No: 15, Galle Road, Kandy'),
    ('C003', 'Saman', 'Rathnayake', NULL, '0784567890', 'Person', 'No: 5, Hill Road, Matara'),
    ('C004', 'Kumari', 'Dissanayake', NULL , '0756789012', 'Person', 'No: 8, Lake View, Nuwara Eliya'),
    ('C005', NULL, NULL, 'Elite Construction Solutions','0763456789', 'Company','No: 20, High Street, Galle'),
    ('C006', NULL, NULL, 'Global Builders Inc', '0725678901', 'Company','No: 25,Beach Road, Negombo'),
    ('C007', NULL, NULL, 'Silverline Developer', '0797890123','Company','No: 12, Temple Road, Anuradhapura')
;
```

- INSERT INTO customer(customer_id, FName, LName, company_name, Telephone, Type, Addresse): This part specifies the table (customer) and the columns (customer_id, FName, LName, company_name, Telephone, Type, Addresse) into which the data will be inserted.
- VALUES: This keyword is followed by a list of comma-separated value sets, each enclosed within parentheses. Each value set represents the data to be inserted into one row of the table.

- Data insert into project table

```sql
INSERT INTO project (Project_ID, Customer_ID, Site_location, Actual_Completion_Date, Start_Date, Estimated_Completion_Date, week_no,
Project_Type, Specific_Project_Requirement, Current_Progress)
VALUES
('P001', 'C001', 'Negombo', '2023-01-21', '2023-01-01', '2023-01-20', 3, 'Renovation', 'Renovating a beach resort', 'Completed'),
('P002', 'C002', 'Kalutara', '2023-07-01', '2023-06-01', '2023-07-01', 4, 'Renovation', 'Renovating a temple', 'Completed'),
('P003', 'C003', 'Gampaha', '2023-10-26', '2023-10-12', '2023-10-26', 2, 'Renovation', 'Building a housing scheme', 'Completed'),
('P004', 'C004', 'Batticaloa', '2023-05-22', '2023-04-01', '2023-05-22', 7, 'Construction', 'Constructing a hospital', 'Completed'),
('P005', 'C005', 'Polonnaruwa', '2023-08-27', '2023-08-06', '2023-08-20', 3, 'Construction', 'Renovating an ancient palace', 'Completed'),
('P006', 'C006', 'Gampaha', '2024-01-20', '2023-12-13', '2024-01-20', 6, 'Construction', 'Building a commercial complex', 'Completed'),
('P007', 'C007', 'Gampaha', NULL, '2024-03-10', NULL, 3, 'Renovation', 'Renovating a shopping mall', 'Planning'),
('P008', 'C001', 'Kandy', NULL, '2024-03-14', NULL, 2, 'Construction', 'Building a hotel', 'Planning'),
('P009', 'C002', 'Matara', NULL, '2024-03-25', NULL, 1, 'Renovation', 'Renovating a historical building', 'Planning'),
('P010', 'C003', 'Kandy', NULL, '2024-02-10', '2024-04-20', 10, 'Construction', 'Building a shopping complex', 'Ongoing'),
('P011', 'C004', 'Kandy', NULL, '2024-02-20', '2024-04-03', 6, 'Renovation', 'Renovating a heritage hotel', 'Ongoing'),
('P012', 'C001', 'Gampaha', NULL, '2024-03-12', '2024-05-07', 8, 'Construction', 'Building a commercial tower', 'Ongoing'),
('P013', 'C002', 'Kandy', NULL, '2024-03-20', '2024-04-17', 4, 'Renovation', 'Renovating a historical temple', 'Ongoing'),
('P014', 'C003', 'Gampaha', NULL, '2024-02-25', '2024-04-28', 9, 'Construction', 'Building a resort', 'Ongoing'),
('P015', 'C004', 'Kandy', NULL, '2024-03-17', '2024-05-12', 8, 'Construction', 'Constructing a hospital', 'Ongoing');
```

- INSERT INTO project (Project_ID, Customer_ID, Site_location, Actual_Completion_Date, Start_Date, Estimated_Completion_Date, Week_No, Project_Type, Specific_Project_Requirement, Current_Progress): This part specifies the table (project) and the columns into which the data will be inserted.
- VALUES: This keyword is followed by a list of comma-separated value sets, each enclosed within parentheses. Each value set represents the data to be inserted into one row of the table.

- Data insert into staff table

```sql
INSERT INTO Staff (Staff_ID, Salary, FName, LName, Role, Telephone, Email)
VALUES
    ('S001', 75000, 'Chaminda', 'Silva', 'Construction Worker', '077-1234567', 'chaminda_silva@example.com'),
    ('S002', 80000, 'Saman', 'Fernando', 'Construction Worker', '077-2345678', 'saman_fernando@example.com'),
    ('S003', 60000, 'Kamal', 'Perera', 'Construction Worker', '071-3456789', 'kamal_perera@example.com'),
    ('S004', 70000, 'Anula', 'Jayawardena', 'Construction Worker', '076-4567890', 'anula_jayawardena@example.com'),
    ('S005', 90000, 'Roshan', 'Silva', 'Construction Worker', '070-5678901', 'roshan_silva@example.com'),
    ('S006', 270000, 'Ruwan', 'Perera', 'Supervisor', '070-0123456', 'ruwan_perera@example.com'),
    ('S007', 250000, 'Udara', 'Jayasuriya', 'Supervisor', '072-5678901', 'udara_jayasuriya@example.com'),
    ('S008', 400000, 'Lakshmi', 'Rathnayake', 'Supervisor', '079-7890123', 'lakshmi_rathnayake@example.com'),
    ('S009', 380000, 'Sunil', 'Perera', 'Supervisor', '073-8901234', 'sunil_perera@example.com'),
    ('S010', 420000, 'Sanjeewa', 'Fernando', 'Supervisor', '074-9012345', 'sanjeewa_fernando@example.com')
;
```

- INSERT INTO Staff (Staff_ID, Salary, FName, LName, Role, Telephone, Email): This part specifies the table (Staff) and the columns into which the data will be inserted.
- VALUES: This keyword is followed by a list of comma-separated value sets, each enclosed within parentheses. Each value set represents the data to be inserted into one row of the table.
- Each value set contains values for the columns specified in the INSERT INTO statement, in the same order.

- Data insert to supervisor table

```
INSERT INTO Supervisor (Staff_ID, Allowance, Supisor_ID, Privillages)
VALUES
('S006',67000,'SP01','Car'),
('S007',55000,'SP02','Petrol'),
('S008', 380000, 'SP03', 'Authorization to approve project changes'),
('S009', 400000, 'SP04', 'Full access to project resources'),
('S010', 400000, 'SP05', 'Full access to project resources')
;
```

- INSERT INTO Supervisor (Staff_ID, Allowance, Supervisor_ID, Privileges): This part specifies the table Supervisor and the columns into which the data will be inserted.

- VALUES: This keyword is followed by a list of comma-separated value sets, each enclosed within parentheses. Each value set represents the data to be inserted into one row of the table.

- Each value set contains values for the columns specified in the INSERT INTO statement, in the same order.

Data insert into payment installment table

```
INSERT INTO payment_instalement_plan
(instalement_id, project_id, customer_id, payment_methoad, type_of_payment, full_amount, status_,
number_of_instalemnt)
VALUES
('I001' , 'P004', 'C004', 'Cash','Instalement payment', 400000,  'Finshed', 4 ),
('I002' , 'P005', 'C005',  'Credit Card','Instalement payment', 500000, 'Finshed', 5 ),
('0003' , 'P006', 'C006', 'Cash','Instalement payment', 1000000, 'Finshed', 10 ),
('I004' , 'P007', 'C007', 'Bank Transfer','Instalement payment', 20000000, 'Pending', 20 ),
('I005' , 'P011', 'C005', 'Bank Transfer','Instalement payment', 3000000, 'Pending', 30 ),
('I006' , 'P012', 'C001', 'Bank Transfer','Instalement payment', 200000, 'Pending', 20),
('I007' , 'P013', 'C002', 'Bank Transfer','Instalement payment', 800000, 'Finshed', 80 );
```

- INSERT INTO payment_instalment_plan: This part specifies the table payment_instalment_plan into which the data will be inserted.
- (instalment_id, project_id, customer_id, payment_method, type_of_payment, full_amount, status_, number_of_instalment): This part specifies the columns into which the data will be inserted.
- VALUES: This keyword is followed by a list of comma-separated value sets, each enclosed within parentheses. Each value set represents the data to be inserted into one row of the table.
- Each value set contains values for the columns specified in the INSERT INTO statement, in the same order.

- Data insert into payment one time table

```
INSERT INTO payment_one_time_plan(one_time_id, project_id, customer_id,type_of_payment ,payment_methoad,
full_amount, status_, discount_precentage)
VALUES

('F001', 'P001', 'C001','One time','Cash', 50000000, 'Finshed', '10%'),
('F002', 'P002', 'C002','One time','Bank Transfer', 400000, 'Finshed', '8%'),
('F003','P003', 'C003','One time','Cash', 200000, 'Finshed', '3%'),
('F004','P008', 'C007','One time','Bank Transfer', 10000000, 'Pending', '4%' ),
('F005','P009', 'C001','One time','Cash', 100000, 'Pending', '2%'),
('F006','P010', 'C003','One time','Bank Transfer', 500000, 'Pending', '3.5%'),
('F007','P014', 'C003','Cash','Bank Transfer', 800000, 'Finshed', '4.2%'),
('F008','P015', 'C004','One time','Bank Transfer', 300000, 'Finshed', '2.4%' );
```

- INSERT INTO payment_one_time_plan: This part specifies the table payment_one_time_plan into which the data will be inserted.
- (one_time_id, project_id, customer_id, type_of_payment, payment_method, full_amount, status_, discount_percentage): This part specifies the columns into which the data will be inserted.
- VALUES: This keyword is followed by a list of comma-separated value sets, each enclosed within parentheses. Each value set represents the data to be inserted into one row of the table.
- Each value set contains values for the columns specified in the INSERT INTO statement, in the same order.

- Data insert into duty assignment table

```
INSERT INTO duty_assignment(duty_assignment_id, staff_id, project_id, current_week_no, hours_worked)
VALUES
("T001","S001","P001",1,2),     ("T002","S002","P001",1,3),     ("T003","S003","P001",1,2),
("T004","S004","P001",1,3),     ("T005","S005","P001",1,3),     ("T006","S006","P001",1,3),
("T007","S007","P001",1,3),     ("T008","S008","P001",1,2),     ("T009","S009","P001",1,2),
("T010","S010","P001",1,4),     ("T011","S001","P001",2,4),     ("T012","S002","P001",2,3),
("T013","S003","P001",2,3),     ("T014","S004","P001",2,4),     ("T015","S005","P001",2,4),
("T016","S006","P001",2,4),     ("T017","S007","P001",2,4),     ("T018","S008","P001",2,4),
("T019","S009","P001",2,3),     ("T020","S010","P001",2,3),     ("T021","S001","P001",3,2),
("T022","S002","P001",3,2),     ("T023","S003","P001",3,2),     ("T024","S004","P001",3,1),
("T025","S005","P001",3,1),     ("T026","S006","P001",3,1),     ("T027","S007","P001",3,1),
("T028","S008","P001",3,1),     ("T029","S009","P001",3,1),     ("T030","S010","P001",3,1),
("T031","S001","P002",1,1),     ("T032","S002","P002",1,3),     ("T033","S003","P002",1,3),
("T034","S004","P002",1,3),     ("T035","S005","P002",1,4),     ("T036","S006","P002",1,4),
("T037","S007","P002",1,1),     ("T038","S008","P002",1,2),     ("T039","S009","P002",1,3),
("T040","S010","P002",1,4),     ("T041","S001","P002",2,5),     ("T042","S002","P002",2,5),
("T043","S003","P002",2,4),     ("T044","S004","P002",2,3),
("T045","S005","P002",2,2),     ("T046","S006","P002",2,3),
("T047","S007","P002",2,4),     ("T048","S008","P002",2,5),
("T049","S009","P002",2,6),     ("T050","S010","P002",2,3);
```

```
INSERT INTO duty_assignment(duty_assignment_id, staff_id, project_id, current_week_no, hours_worked)
VALUES
("T051","S001","P002",3,4),("T052","S002","P002",3,2),("T053","S003","P002",3,3),("T054","S004","P002",3,4),
("T055","S005","P002",3,1),("T056","S006","P002",3,4),("T057","S007","P002",3,4),("T058","S008","P002",3,4),
("T059","S009","P002",3,4),("T060","S010","P002",3,4),("T061","S001","P002",4,4),("T062","S002","P002",4,4),
("T063","S003","P002",4,4),("T064","S004","P002",4,4),("T065","S005","P002",4,4),("T066","S006","P002",4,4),
("T067","S007","P002",4,2),("T068","S008","P002",4,3),("T069","S009","P002",4,2),("T070","S010","P002",4,3),
("T071","S001","P003",1,3),("T072","S002","P003",1,3),("T073","S003","P003",1,3),("T074","S004","P003",1,2),
("T075","S005","P003",1,2),("T076","S006","P003",1,4),("T077","S007","P003",1,4),("T078","S008","P003",1,3),
("T079","S009","P003",1,3),("T080","S010","P003",1,4),("T081","S001","P003",2,4),("T082","S002","P003",2,4),
("T083","S003","P003",2,4),("T084","S004","P003",2,4),("T085","S005","P003",2,3),("T086","S006","P003",2,3),
("T087","S007","P003",2,2),("T088","S008","P003",2,2),("T089","S009","P003",2,2),("T090","S010","P003",2,1),
("T091","S001","P004",1,1),("T092","S002","P004",1,1),("T093","S003","P004",1,1),("T094","S004","P004",1,1),
("T095","S005","P004",1,1),("T096","S006","P004",1,1),("T097","S007","P004",1,1),("T098","S008","P004",1,3),
("T099","S009","P004",1,3),("T100","S010","P004",1,3),("T101","S001","P004",2,4),("T102","S002","P004",2,4),
("T103","S003","P004",2,1),("T104","S004","P004",2,2),("T105","S005","P004",2,3),("T106","S006","P004",2,4),
("T107","S007","P004",2,5),("T108","S008","P004",2,5),("T109","S009","P004",2,4),("T110","S010","P004",2,3),
("T111","S001","P004",3,2),("T112","S002","P004",3,3),("T113","S003","P004",3,4),("T114","S004","P004",3,5),
("T115","S005","P004",3,6),("T116","S006","P004",3,3),("T117","S007","P004",3,4),("T118","S008","P004",3,2),
("T119","S009","P004",3,3),("T120","S010","P004",3,4),("T121","S001","P004",4,1),("T122","S002","P004",4,4),
("T123","S003","P004",4,4),("T124","S004","P004",4,4),("T125","S005","P004",4,4),("T126","S006","P004",4,4),
("T127","S007","P004",4,4),("T128","S008","P004",4,4),("T129","S009","P004",4,4),("T130","S010","P004",4,4),
("T131","S001","P004",5,4),("T132","S002","P004",5,4),("T133","S003","P004",5,2),("T134","S004","P004",5,3),
("T135","S005","P004",5,2),("T136","S006","P004",5,3),("T137","S007","P004",5,3),("T138","S008","P004",5,3),
("T139","S009","P004",5,3),("T140","S010","P004",5,2),("T141","S001","P004",6,2),("T142","S002","P004",6,4),
("T143","S003","P004",6,4),("T144","S004","P004",6,3),("T145","S005","P004",6,3),("T146","S006","P004",6,4),
("T147","S007","P004",6,4),("T148","S008","P004",6,4),("T149","S009","P004",6,4),("T150","S010","P004",6,4);
```

```
INSERT INTO duty_assignment(duty_assignment_id, staff_id, project_id, current_week_no, hours_worked)
VALUES

("T401","S001","P010",10,3),("T402","S002","P010",10,4),("T403","S003","P010",10,5),("T404","S004","P010",10,6),("T405","S005","P010",10,3),("T406","S006","P010",10,4),("T407","S007","P010",10,2),("T408","S008","P010",10,3),("T409","S009","P010",10,4),("T410","S010","P010",10,1),
("T411","S001","P011",1,4),("T412","S002","P011",1,4),("T413","S003","P011",1,4),("T414","S004","P011",1,4),("T415","S005","P011",1,4),("T416","S006","P011",1,4),("T417","S007","P011",1,4),("T418","S008","P011",1,4),("T419","S009","P011",1,4),("T420","S010","P011",1,4),
("T421","S001","P011",2,4),("T422","S002","P011",2,5),("T423","S003","P011",2,5),("T424","S004","P011",2,5),("T425","S005","P011",2,5),("T426","S006","P011",2,5),("T427","S007","P011",2,5),("T428","S008","P011",2,5),("T429","S009","P011",2,5),("T430","S010","P011",2,5),
("T431","S001","P011",3,5),("T432","S002","P011",3,5),("T433","S003","P011",3,5),("T434","S004","P011",3,5),("T435","S005","P011",3,5),("T436","S006","P011",3,5),("T437","S007","P011",3,5),("T438","S008","P011",3,5),("T439","S009","P011",3,5),("T440","S010","P011",3,5),
("T441","S001","P011",4,5),("T442","S002","P011",4,5),("T443","S003","P011",4,4),("T444","S004","P011",4,4),("T445","S005","P011",4,4),("T446","S006","P011",4,4),("T447","S007","P011",4,1),("T448","S008","P011",4,1),("T449","S009","P011",4,1),("T450","S010","P011",4,1),
("T451","S001","P011",5,1),("T452","S002","P011",5,1),("T453","S003","P011",5,1),("T454","S004","P011",5,1),("T455","S005","P011",5,1),("T456","S006","P011",5,1),("T457","S007","P011",5,1),("T458","S008","P011",5,1),("T459","S009","P011",5,2),("T460","S010","P011",5,3),
("T461","S001","P011",6,2),("T462","S002","P011",6,3),("T463","S003","P011",6,3),("T464","S004","P011",6,3),("T465","S005","P011",6,3),("T466","S006","P011",6,2),("T467","S007","P011",6,2),("T468","S008","P011",6,4),("T469","S009","P011",6,4),("T470","S010","P011",6,3),
("T471","S001","P012",1,3),("T472","S002","P012",1,4),("T473","S003","P012",1,4),("T474","S004","P012",1,4),("T475","S005","P012",1,4),("T476","S006","P012",1,4),("T477","S007","P012",1,3),("T478","S008","P012",1,3),("T479","S009","P012",1,2),("T480","S010","P012",1,2),
("T481","S001","P012",2,2),("T482","S002","P012",2,1),("T483","S003","P012",2,1),("T484","S004","P012",2,1),("T485","S005","P012",2,1),("T486","S006","P012",2,1),("T487","S007","P012",2,1),("T488","S008","P012",2,1),("T489","S009","P012",2,1),("T490","S010","P012",2,3),
("T491","S001","P012",3,3),("T492","S002","P012",3,3),("T493","S003","P012",3,4),("T494","S004","P012",3,4),("T495","S005","P012",3,1),("T496","S006","P012",3,2),("T497","S007","P012",3,3),("T498","S008","P012",3,4),("T499","S009","P012",3,5),("T500","S010","P012",3,5),
("T501","S001","P012",4,3),("T502","S002","P012",4,3),("T503","S003","P012",4,2),("T504","S004","P012",4,3),("T505","S005","P012",4,4),("T506","S006","P012",4,5),("T507","S007","P012",4,6),("T508","S008","P012",4,3),("T509","S009","P012",4,4),("T510","S010","P012",4,2),
("T511","S001","P012",5,3),("T512","S002","P012",5,4),("T513","S003","P012",5,1),("T514","S004","P012",5,4),("T515","S005","P012",5,4),("T516","S006","P012",5,4),("T517","S007","P012",5,4),("T518","S008","P012",5,4),("T519","S009","P012",5,4),("T520","S010","P012",5,4),
("T521","S001","P012",6,4),("T522","S002","P012",6,4),("T523","S003","P012",6,4),("T524","S004","P012",6,4),("T525","S005","P012",6,2),("T526","S006","P012",6,3),("T527","S007","P012",6,2),("T528","S008","P012",6,3),("T529","S009","P012",6,3),("T530","S010","P012",6,3),
("T531","S001","P012",7,3),("T532","S002","P012",7,2),("T533","S003","P012",7,2),("T534","S004","P012",7,4),("T535","S005","P012",7,4),("T536","S006","P012",7,3),("T537","S007","P012",7,3),("T538","S008","P012",7,4),("T539","S009","P012",7,4),("T540","S010","P012",7,4),
("T541","S001","P012",8,4),("T542","S002","P012",8,4),("T543","S003","P012",8,3),("T544","S004","P012",8,3),("T545","S005","P012",8,2),("T546","S006","P012",8,2),("T547","S007","P012",8,2),("T548","S008","P012",8,1),("T549","S009","P012",8,1),("T550","S010","P012",8,1),
("T551","S001","P013",1,1),("T552","S002","P013",1,1),("T553","S003","P013",1,1),("T554","S004","P013",1,1),("T555","S005","P013",1,1),("T556","S006","P013",1,3),("T557","S007","P013",1,3),("T558","S008","P013",1,3),("T559","S009","P013",1,4),("T560","S010","P013",1,4),
("T561","S001","P013",2,1),("T562","S002","P013",2,2),("T563","S003","P013",2,3),("T564","S004","P013",2,4),("T565","S005","P013",2,5),("T566","S006","P013",2,5),("T567","S007","P013",2,4),("T568","S008","P013",2,3),("T569","S009","P013",2,2),("T570","S010","P013",2,3),
("T571","S001","P013",3,4),("T572","S002","P013",3,5),("T573","S003","P013",3,6),("T574","S004","P013",3,3),("T575","S005","P013",3,4),("T576","S006","P013",3,2),("T577","S007","P013",3,3),("T578","S008","P013",3,4),("T579","S009","P013",3,1),("T580","S010","P013",3,4),
("T581","S001","P013",4,4),("T582","S002","P013",4,4),("T583","S003","P013",4,4),("T584","S004","P013",4,4),("T585","S005","P013",4,4),("T586","S006","P013",4,4),("T587","S007","P013",4,4),("T588","S008","P013",4,4),("T589","S009","P013",4,4),("T590","S010","P013",4,4),
("T591","S001","P014",1,2),("T592","S002","P014",1,3),("T593","S003","P014",1,2),("T594","S004","P014",1,3),("T595","S005","P014",1,3),("T596","S006","P014",1,3),("T597","S007","P014",1,3),("T598","S008","P014",1,2),("T599","S009","P014",1,2),("T600","S010","P014",1,4),
("T601","S001","P014",2,4),("T602","S002","P014",2,3),("T603","S003","P014",2,3),("T604","S004","P014",2,4),("T605","S005","P014",2,4),("T606","S006","P014",2,4),("T607","S007","P014",2,4),("T608","S008","P014",2,4),("T609","S009","P014",2,3),("T610","S010","P014",2,3),
("T611","S001","P014",3,2),("T612","S002","P014",3,2),("T613","S003","P014",3,2),("T614","S004","P014",3,1),("T615","S005","P014",3,1),("T616","S006","P014",3,1),("T617","S007","P014",3,1),("T618","S008","P014",3,1),("T619","S009","P014",3,1),("T620","S010","P014",3,1),
("T621","S001","P014",4,1),("T622","S002","P014",4,3),("T623","S003","P014",4,3),("T624","S004","P014",4,3),("T625","S005","P014",4,4),("T626","S006","P014",4,4),("T627","S007","P014",4,1),("T628","S008","P014",4,2),("T629","S009","P014",4,3),("T630","S010","P014",4,4),
("T631","S001","P014",5,5),("T632","S002","P014",5,5),("T633","S003","P014",5,4),("T634","S004","P014",5,3),("T635","S005","P014",5,2),("T636","S006","P014",5,3),("T637","S007","P014",5,4),("T638","S008","P014",5,5),("T639","S009","P014",5,6),("T640","S010","P014",5,3),
("T641","S001","P014",6,4),("T642","S002","P014",6,2),("T643","S003","P014",6,3),("T644","S004","P014",6,4),("T645","S005","P014",6,1),("T646","S006","P014",6,4),("T647","S007","P014",6,4),("T648","S008","P014",6,4),("T649","S009","P014",6,4),("T650","S010","P014",6,4),
("T651","S001","P014",7,4),("T652","S002","P014",7,4),("T653","S003","P014",7,4),("T654","S004","P014",7,4),("T655","S005","P014",7,4),("T656","S006","P014",7,4),("T657","S007","P014",7,5),("T658","S008","P014",7,5),("T659","S009","P014",7,5),("T660","S010","P014",7,5),
("T661","S001","P014",8,5),("T662","S002","P014",8,5),("T663","S003","P014",8,5),("T664","S004","P014",8,5),("T665","S005","P014",8,5),("T666","S006","P014",8,5),("T667","S007","P014",8,5),("T668","S008","P014",8,5),("T669","S009","P014",8,5),("T670","S010","P014",8,5),
("T671","S001","P014",9,5),("T672","S002","P014",9,5),("T673","S003","P014",9,5),("T674","S004","P014",9,5),("T675","S005","P014",9,5),("T676","S006","P014",9,5),("T677","S007","P014",9,5),("T678","S008","P014",9,4),("T679","S009","P014",9,4),("T680","S010","P014",9,4),
("T681","S001","P015",1,4),("T682","S002","P015",1,5),("T683","S003","P015",1,5),("T684","S004","P015",1,5),("T685","S005","P015",1,5),("T686","S006","P015",1,5),("T687","S007","P015",1,5),("T688","S008","P015",1,5),("T689","S009","P015",1,5),("T690","S010","P015",1,5),
("T691","S001","P015",2,5),("T692","S002","P015",2,2),("T693","S003","P015",2,3),("T694","S004","P015",2,2),("T695","S005","P015",2,3),("T696","S006","P015",2,3),("T697","S007","P015",2,3),("T698","S008","P015",2,3),("T699","S009","P015",2,2),("T700","S010","P015",2,2);
```

- matching the respective column specified in the INSERT INTO statement.
- This statement inserts multiple records into the duty_assignment table. Each VALUES clause represents a set of values for one record to be inserted, and the values correspond to the columns specified in the INSERT INTO statement. Each set of values corresponds to one row in the duty_assignment table, with each value

Section 4:

data manipulation with SQL

a).

```
SELECT *
FROM project
WHERE (site_location = 'Colombo' OR site_location = 'Kandy')
    AND actual_completion_date BETWEEN
DATE_SUB(CURRENT_DATE(), INTERVAL 6 MONTH) AND
CURRENT_DATE()
    AND current_progress = 'Completed';
```

What I asked:

```
1 SELECT *
2 FROM project
3 WHERE (site_location = 'colombo' OR site_location = 'Kandy')
4 AND (actual_completion_date >= DATE_SUB(CURRENT_DATE(), INTERVAL 6 MONTH));
5
```

What system given:

Show query box

Showing rows 0 - 1 (2 total, Query took 0.0005 seconds.)

```
SELECT * FROM project WHERE (site_location = 'colombo' OR site_location = 'Kandy') AND (actual_completion_date >= DATE_SUB(CURRENT_DATE(), INTERVAL 6 MONTH));
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	project_id	customer_id	site_location	week_no	actual_completion_date	start_date	estimated_completion_date	project_type	specific_project_requirement	current_progress	payment_type
Edit Copy Delete	P003	C003	Colombo	2	2023-10-26	2023-10-12	2023-10-26	Renovation	Building a housing scheme	completed	One time
Edit Copy Delete	P006	C006	Kandy	6	2024-01-20	2023-12-13	2024-01-20	Construction	Building a commercial complex	completed	instalement payment

Check all With selected: Edit Copy Delete Export

project_id	customer_id	site_location	week_no	actual_completion_date	start_date	estima
P003	C003	Colombo	2	2023-10-26	2023-10-12	2023-1
P006	C006	Kandy	6	2024-01-20	2023-12-13	2024-0

current_progress	p
completed	O
completed	in pa

b).

```
SELECT
    project.project_id,
    customer.company_name,
    project.estimated_completion_date,
    SUM(payment_one_time_plan.full_amount) AS total_amount,
    SUM(payment_instalement_plan.sum_payment_currently ) AS
total_amount
FROM
    project
INNER JOIN
    customer ON project.customer_id = customer.customer_id
LEFT JOIN
    payment_one_time_plan ON payment_one_time_plan.project_id =
project.project_id
LEFT JOIN
    payment_instalement_plan ON payment_instalement_plan.project_id =
project.project_id

WHERE
    project.current_progress = 'ongoing' AND
    project.project_type = 'renovation' AND
    customer.type = 'company'
GROUP BY
    project.project_id,
    customer.company_name,
    project.estimated_completion_date;
```

what I asked:

```sql
SELECT
    project.project_id,
    customer.company_name,
    project.estimated_completion_date,
    SUM(payment_one_time_plan.full_amount) AS total_amount,
    SUM(payment_instalement_plan.sum_payment_currently ) AS total_amount
FROM
    project
INNER JOIN
    customer ON project.customer_id = customer.customer_id
LEFT JOIN
    payment_one_time_plan ON payment_one_time_plan.project_id = project.project_id
LEFT JOIN
    payment_instalement_plan ON payment_instalement_plan.project_id = project.project_id

WHERE
    project.current_progress = 'ongoing' AND
    project.project_type = 'renovation' AND
    customer.type = 'company'
GROUP BY
    project.project_id,
    customer.company_name,
    project.estimated_completion_date;
```

What system given:

project_id	company_name	estimated_completion_date	total_amount	total_amount
P007	Silverline Developer	2024-06-10	*NULL*	8000000
P016	Evergreen Landscaping Services	2024-04-15	100000	*NULL*

The end.