Allyson Totpal

CIS-535

January 19,2025

Dear Bob,

I appreciate the opportunity to work on developing a database solution for your business, Bob’s Home Repairs. Currently your business operations are manually tracked, which can lead to limited business growth due to how time-consuming it can be. Based on your business outline and analysis of business operations, I have designed a structured and efficient database system to help you streamline business processes, improve record-keeping, and enhance productivity.

Key Features:

1. Customer & Bids Management - A structured way to store customer information and track bids for home repairs where each bid is associated with a customer, allowing for easy retrieval and management.

2. Repair Tracking - Once a bid has been accepted, repairs associated with that big are logged into the system to make it easy to monitor their progress.

3. Material & Supplier Management - The database will maintain records of materials used for each repair and their respective suppliers to help track inventory, costs, and supplier relationships.

4. Payment Tracking - The system enables tracking of payments made by customers to ensure an organized billing process. A similar payment system is also recorded for suppliers.

Deliverables

- Entity-Relationship Diagram (ERD) to outline the structure of the database, detailing entities and their relationships.

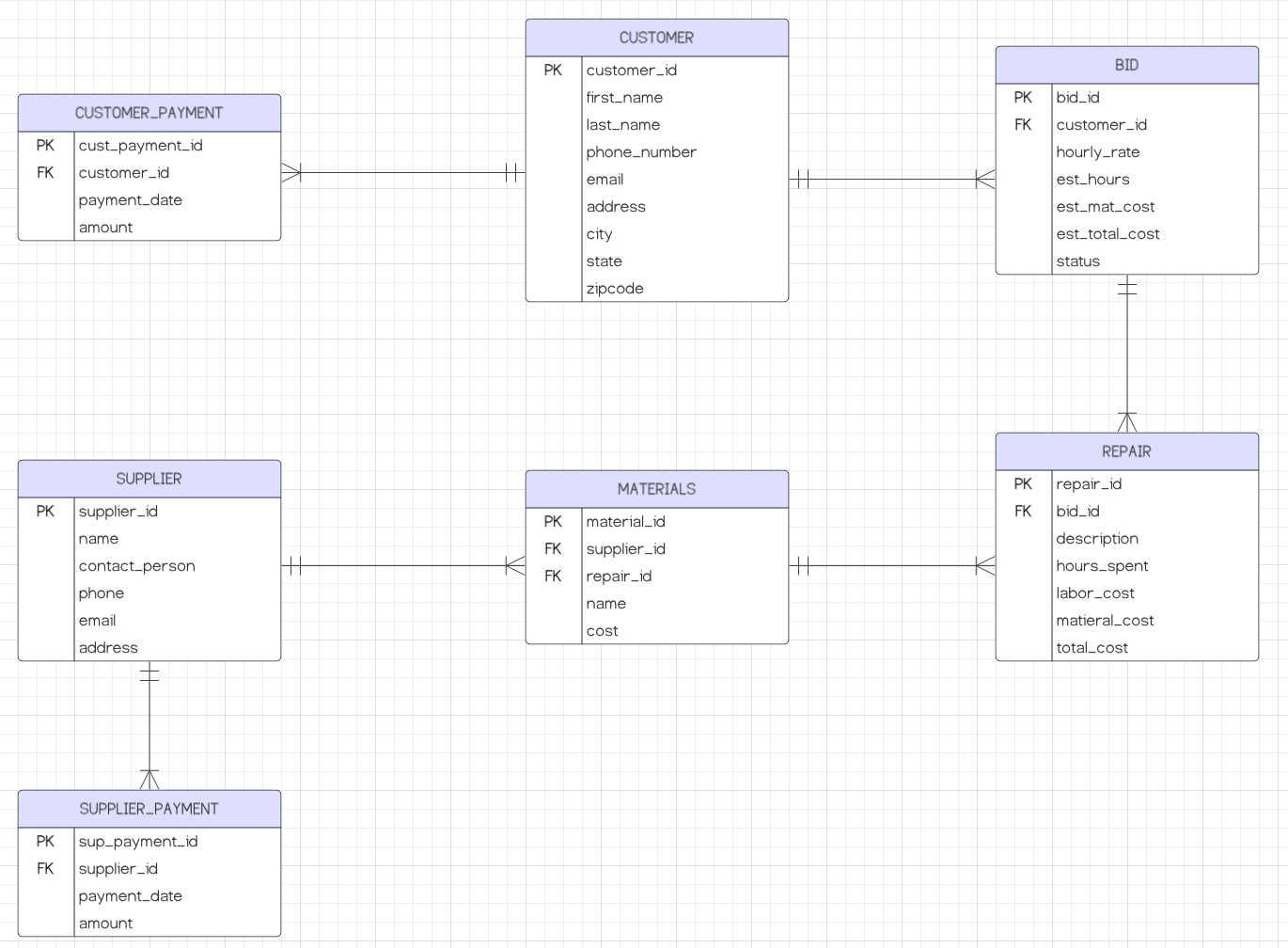
- Third normal form ERD to ensure efficient data organization and integrity, with necessary primary and foreign keys established.

- SQL queries to retrieve essential business information efficiently to help with decision-making and record-keeping.

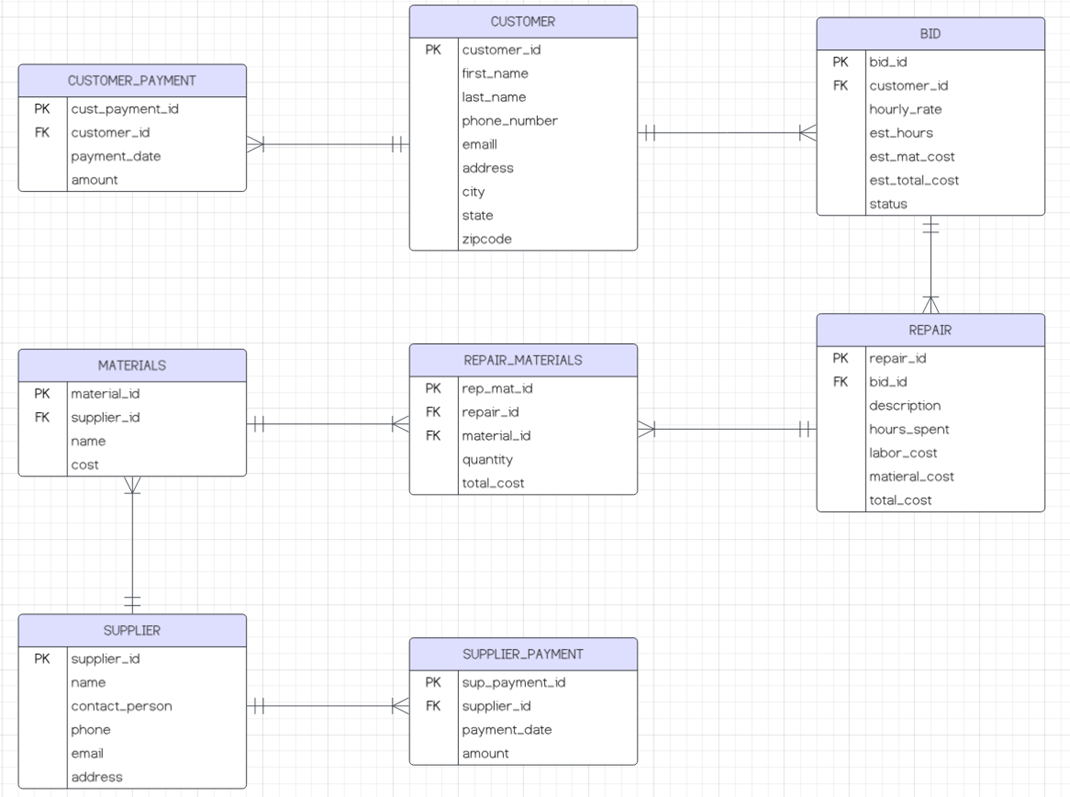
This software package consolidates all project elements into a structured document, making it easy for you to review and use. The benefits this can bring to your business include eliminating the need for manual bookkeeping to reduce errors and save you time, providing a structured way to store and retrieve business data efficiently, enhancing financial tracking for both customer and supplier payments, and scalability of your business to allow for you to manage more customers and repairs jobs seamlessly.

Thank you for the opportunity to contribute to your business’s success and I look forward to your feedback and assisting with implementation as needed.

**ERD**



**Third Normal Form ERD**

****

**DDL - Creation of Tables**

create table Customers (

customer\_id int primary key,

first\_name varchar(30) not null,

last\_name varchar(30),

phone varchar(20),

email varchar(50),

address text

)

create table Bids (

bid\_id int primary key,

customer\_id int foreign key references Customers(customer\_id),

hourly\_rate float not null,

est\_hours float not null,

est\_mat\_cost float not null,

est\_total\_cost float not null,

status varchar(50) not null

)

create table Repairs (

repair\_id int primary key,

bid\_id int foreign key references Bids(bid\_id),

description text not null,

hours\_spent float,

labor\_cost float,

material\_cost float,

total\_cost float

)

create table Suppliers (

supplier\_id int primary key,

supplier\_name varchar(100) not null,

contact\_name varchar(50),

phone varchar(20),

email varchar(50),

address text

)

create table Materials (

material\_id int primary key,

supplier\_id int foreign key references Suppliers(supplier\_id),

material\_name varchar(200) not null,

cost float not null

)

create table RepairMaterials (

rep\_mat\_id int primary key,

repair\_id int foreign key references Repairs(repair\_id),

material\_id int foreign key references Materials(material\_id),

quantity int not null,

cost float not null

)

create table SupplierPayments (

sup\_payment\_id int primary key,

supplier\_id int foreign key references Suppliers(supplier\_id),

payment\_date date not null,

amount float not null

)

create table CustomerPayments (

cust\_payment\_id int primary key,

customer\_id int foreign key references Customers(customer\_id),

payment\_date date not null,

amount float not null

)

**DML - Inserting Records**

insert into Customers(customer\_id, first\_name, last\_name, phone, email, address) values

(1, 'Jane', 'Wozowski','9485559386', 'jane.wozowski@ex.com', '104 Elm Street, Anaheim, CA 92807'),

(2, 'Mike', 'Sulley', '71455510234', 'mike.sulley@ex.com', '238 Oak St, Anaheim, CA 92812'),

(3, 'Elsie', 'Silver', '7145552856', 'elsie.silver@ex.com', '9238 Pine Avenue, Anaheim, CA 92804'),

(4, 'Lyla', 'Sage', '7145552395', 'lyla.sage@ex.com', '7844 Acacia Drive, Anaheim, CA 92817'),

(5, 'Tom', 'Matson', '7145556235', 'tom.matson@ex.com', '29345 Maple Ave, Anaheim, CA 92801')

insert into Bids(bid\_id, customer\_id, hourly\_rate, est\_hours, est\_mat\_cost, est\_total\_cost, status) values

(1, 1, 50.00, 10, 200.00, 700.00, 'Pending'),

(2, 2, 55.00, 8, 150.00, 590.00, 'Accepted'),

(3, 3, 60.00, 12, 300.00, 1020.00, 'Rejected'),

(4, 4, 50.00, 5, 100.00, 350.00, 'Completed'),

(5, 5, 65.00, 7, 250.00, 705.00, 'Accepted')

insert into Repairs(repair\_id, bid\_id, description, hours\_spent, labor\_cost, material\_cost, total\_cost) values

(1, 1, 'Fix plumbing leak', 10, 500.00, 200.00, 700.00),

(2, 2, 'Install new roof', 8, 440.00, 150.00, 590.00),

(3, 3, 'Paint exterior', 12, 720.00, 300.00, 1020.00),

(4, 4, 'Replace flooring', 5, 250.00, 100.00, 350.00),

(5, 5, 'Repair foundation', 7, 455.00, 250.00, 705.00)

insert into Suppliers(supplier\_id, supplier\_name, contact\_name, phone, email, address) values

(1, 'Home Depot', 'Alice Brown', '7145551239', 'alice@homedepot.com', '101 Main St'),

(2, 'Lowes', 'Bob Green', '7145553284', 'bob@lowes.com', '983 South St'),

(3, 'Ace Hardware', 'Charlie White', '7145559123', 'charlie@acehardware.com', '1430 East St'),

(4, 'Ganahl Lumber', 'David Black', '7145555648', 'david@ganahllumber.com', '986 West St'),

(5, 'Harbor Freight', 'Eve Red', '7145553291', 'eve@harborfreight.com', '55 North St')

insert into Materials(material\_id, supplier\_id, material\_name, cost) values

(1, 1, 'Nails', 5.00),

(2, 1, 'Lumber', 20.00),

(3, 2, 'Paint', 30.00),

(4, 3, 'Tiles', 25.00),

(5, 4, 'Cement', 15.00),

(6, 5, 'Screws', 6.00)

insert into RepairMaterials(rep\_mat\_id, repair\_id, material\_id, quantity, cost) values

(1, 1, 1, 10, 50.00),

(2, 2, 2, 5, 100.00),

(3, 3, 3, 3, 90.00),

(4, 4, 4, 4, 100.00),

(5, 5, 5, 6, 90.00)

insert into SupplierPayments(sup\_payment\_id, supplier\_id, payment\_date, amount) values

(1, 1, '2025-01-01', 200.00),

(2, 2, '2025-01-05', 150.00),

(3, 3, '2025-01-10', 300.00),

(4, 4, '2025-01-15', 100.00),

(5, 5, '2025-01-20', 250.00)

insert into CustomerPayments(cust\_payment\_id, customer\_id, payment\_date, amount) values

(1, 1, '2025-01-21', 100.00),

(2, 2, '2025-01-23', 200.00),

(3, 3, '2024-01-25', 150.00),

(4, 4, '2024-01-27', 300.00),

(5, 5, '2024-01-29', 250.00)

**SQL Queries**

-- List the names of the customers who provided bids last month.

select c.first\_name, c.last\_name

from Customers c

join Bids b on c.customer\_id = b.customer\_id

where b.bid\_date >=dateadd(month, -1, datefromparts(year(getdate()), month(getdate()), 1))

and b.bid\_date < datefromparts(year(getdate()), month(getdate()), 1)

-- List all the customers that have an outstanding balance (amount due to Bob) that is greater than $500.

select c.first\_name, c.last\_name, (coalesce(sum(r.total\_cost), 0) - coalesce(sum(cp.amount), 0)) as outstanding\_balance

from Customers c

join Bids b on c.customer\_id = b.customer\_id

join Repairs r on b.bid\_id = r.bid\_id

left join CustomerPayments cp on c.customer\_id = cp.customer\_id

group by c.customer\_id, c.first\_name, c.last\_name

having (coalesce(sum(r.labor\_cost + r.material\_cost), 0) - coalesce(sum(cp.amount), 0)) > 500

-- List all the suppliers where Bob has an outstanding balance that is greater than $1,000.

select s.supplier\_name as supplier\_name, (coalesce(sum(m.cost), 0) - coalesce(sum(sp.amount), 0)) as outstanding\_balance

from Suppliers s

join Materials m on s.supplier\_id = m.supplier\_id

left join SupplierPayments sp on s.supplier\_id = sp.supplier\_id

group by s.supplier\_id, s.supplier\_name

having (coalesce(sum(m.cost), 0) - coalesce(sum(sp.amount), 0)) > 1000

-- List all the material that was used for bids where the estimated hours for the bid was greater than 40 hours.

select m.material\_name

from Materials m

join RepairMaterials rm on m.material\_id = rm.material\_id

join Repairs r on rm.repair\_id = r.repair\_id

join Bids b on r.bid\_id = b.bid\_id

where est\_hours > 40

-- List all the material that Bob purchased from a specific supplier (Note: you can decide on the supplier but there should only be one supplier).

select m.material\_name, m.cost

from Materials m

join Suppliers s on m.supplier\_id = s.supplier\_id

where s.supplier\_name = 'Ganahl Lumber'