



BOB'S HOME REPAIRS

Software Package

Megan Eddy

Bellevue University

Management and Design of Databases

Patrick Gold

August 6, 2024

Table of Contents

1. Introduction.....	2
a. Overview of Package.....	2
b. Objective.....	2
2. Entity Relationship Diagram	
a. Description.....	3
b. Entity Relationship Diagram.....	3
3. Normalized Tables.....	4
a. Description.....	4
b. DDL Scripts.....	4
c. DML Scripts.....	5
4. SQL Queries.....	7
a. Description.....	7
b. Queries.....	7

Dear Bob,

Thank you for the opportunity to assist in the digital transformation of your business. It has been a pleasure collaborating with you, and I am confident you will be highly satisfied with the results.

Objective: The primary objective of this project is to transform your manual bookkeeping processes into a streamlined, automated system. By leveraging digital technologies, I aim to enhance efficiency, reduce errors, and reduce manual time and labor for yourself and your etam. This software package is designed to serve as the foundation of your new processes, enhancing your understanding and efficiency. The package includes:

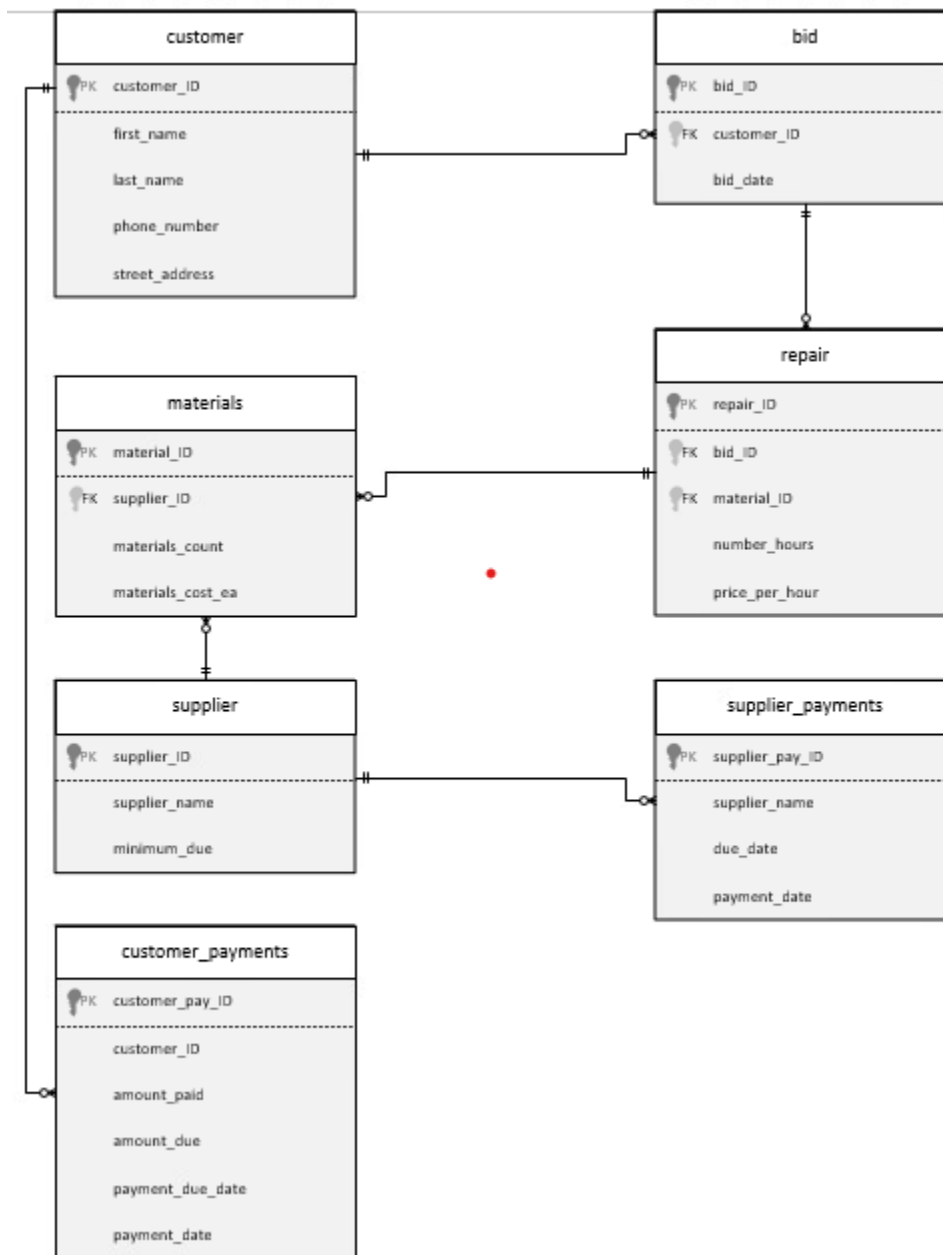
- **Entity-Relationship (E-R) Diagram:** A visual representation of the data structure.
- **Normalized Tables:** Efficiently organized data tables to reduce redundancy and improve data integrity.
- **SQL Scripts:** Pre-written code for creating and managing your database.
- **Query Results:** Sample outputs to demonstrate the effectiveness of the queries.

I hope these materials will facilitate a smooth transition and empower you to leverage your new digital systems effectively.

Entity Relationship (E-R) Diagram

Description

The E-R diagram below forms the basis of the database structure. It was built using the business rules you provided. It efficiently groups different aspects of your business into related tables. Through these relations, data such as customer information, bid information, etc. can be effectively stored and queried without errors. The diagram below shows each table as well as its relationship to other tables. All relationships are one-to-many, maintaining the integrity of the database as a whole.



Normalized Tables

Description

The code below shows the creation of the tables represented in the E-R diagram in the previous section. The SQL DDL Scripts section is the actual table creation while the SQL DML Scripts highlight information input into those tables. Your database consists of seven different tables with each representing a different aspect of your business. Each table has been pre-populated with four rows of example data. The seven tables are:

- Customer – first and last name, phone number and street address
- Bid – customer id, date the bid was received
- Repair – bid id, material id, number of hours, price per hour
- Supplier – supplier name, minimum monthly payment due
- Materials – supplier id, number of materials, materials cost (each), materials description
- Supplier payments – supplier id, material id, due date, payment date
- Customer payments – customer id, amount paid, amount due, due date, payment date

DDL Scripts

```
CREATE TABLE customer (
customer_ID INT PRIMARY KEY,
first_name VARCHAR(25),
last_name VARCHAR(25),
phone_number VARCHAR(20),
street_address VARCHAR(30)
);
```

```
CREATE TABLE bid (
bid_ID INT PRIMARY KEY,
customer_ID INT,
CONSTRAINT fk_customer FOREIGN KEY(customer_ID) REFERENCES customer(customer_ID),
bid_date DATE
);
```

```
CREATE TABLE supplier (
supplier_ID INT PRIMARY KEY,
supplier_name VARCHAR(20),
minimum_due MONEY,
);
```

```
CREATE TABLE materials (
material_ID INT PRIMARY KEY,
supplier_ID INT,
CONSTRAINT fk_supplier FOREIGN KEY(supplier_ID) REFERENCES supplier(supplier_ID),
materials_count INT,
materials_cost_ea MONEY,
materials_descr VARCHAR(30)
);
```

```
CREATE TABLE repair (
repair_ID INT PRIMARY KEY,
bid_ID INT,
CONSTRAINT fk_bid FOREIGN KEY(bid_ID) REFERENCES bid(bid_ID),
```

```
material_ID INT,
CONSTRAINT fk_materials FOREIGN KEY(material_ID) REFERENCES materials(material_ID),
number_hours INT,
price_per_hour MONEY,
);
```

```
CREATE TABLE supplier_payments (
supplier_pay_ID INT PRIMARY KEY,
supplier_ID INT,
CONSTRAINT fk_supplier_2 FOREIGN KEY(supplier_ID) REFERENCES supplier(supplier_ID),
material_ID INT,
CONSTRAINT fk_material_2 FOREIGN KEY(material_ID) REFERENCES materials(material_ID),
due_date DATE,
payment_date DATE
);
```

```
CREATE TABLE customer_payments (
customer_pay_ID INT PRIMARY KEY,
customer_ID INT,
CONSTRAINT fk_customer_2 FOREIGN KEY(customer_ID) REFERENCES customer(customer_ID),
amount_paid MONEY,
amount_due MONEY,
due_date DATE,
payment_date DATE
);
```

DML Scripts

```
INSERT INTO customer VALUES
(1, 'Joseph', 'Dunbar', '540-553-8063', '4123 Jehovah Drive'),
(2, 'Aiden', 'Cockett', '989-649-9246', '306 Mount Street'),
(3, 'Dino', 'Mandic', '503-475-4479', '2052 Kincheloe Road'),
(4, 'Martina', 'Gärtner', '801-954-5352', '3367 Buck Drive'),
(5, 'Elise', 'Kåsa', '484-286-4192', '3028 Renwick Drive');
```

```
INSERT INTO bid VALUES
(1, 5, '2024-06-05'),
(2, 4, '2024-06-10'),
(3, 3, '2024-03-17'),
(4, 2, '2024-07-19'),
(5, 1, '2024-07-04');
```

```
INSERT INTO supplier VALUES
(1, 'Lowe's', 150),
(2, 'Home Depot', 1000),
(3, 'Mr. Fix-It', 500),
(4, 'Ace', 250),
(5, 'Home Store', 875);
```

```
INSERT INTO materials VALUES
(1, 4, 25, 150, 'Planks'),
(2, 5, 50, 75, 'Windows'),
(3, 1, 200, 10, 'Screws'),
(4, 2, 800, 3, 'Doors'),
(5, 3, 100, 100, 'Blinds');
```

```
INSERT INTO repair VALUES
```

```
(1, 1, 1, 25, 300),  
(2, 3, 5, 50, 300),  
(3, 5, 2, 75, 300),  
(4, 4, 4, 100, 300),  
(5, 2, 3, 33, 300);
```

```
INSERT INTO supplier_payments VALUES  
(1, 2, 1, '2024-05-08', '2024-05-07'),  
(2, 3, 5, '2024-05-02', '2024-05-01'),  
(3, 1, 2, '2024-04-02', '2024-04-01'),  
(4, 5, 4, '2024-04-02', '2024-04-01'),  
(5, 4, 3, '2024-04-02', '2024-04-01');
```

```
INSERT INTO customer_payments VALUES  
(1, 1, 200, 100, '2024-07-08', '2024-07-07'),  
(2, 5, 50, 500, '2024-07-02', '2024-06-01'),  
(3, 2, 800, 750, '2024-06-02', '2024-05-01'),  
(4, 4, 75, 50, '2024-06-02', '2024-05-01'),  
(5, 3, 100, 100, '2024-06-02', '2024-05-01');
```

SQL Queries

Description

The queries included below highlight the functionality of the database. They are examples of methods you can use to quickly access timely information about different aspects of your business. This is not an exhaustive list of all queries that can be run, but an example of the types of information that can be gleaned.

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

```
SELECT c.first_name, c.last_name
FROM customer c
JOIN bid b ON c.customer_ID = b.customer_ID
WHERE b.bid_date >= '2024-07-01'
      AND b.bid_date < '2024-08-01';
```

Explanation

Returns the first and last names of every customer who provided a bid in the previous month

2. List all the customers that have an outstanding balance that is greater than \$500

```
SELECT c.first_name, c.last_name
FROM customer c
JOIN customer_payments p ON c.customer_ID = p.customer_ID
WHERE amount_due > 500;
```

Explanation

Returns the first and last names of every customer who have an outstanding balance of more than \$100

3. List all the suppliers owed an outstanding balance that is greater than \$1,000

```
SELECT supplier_name
FROM supplier
WHERE minimum_due > 1000
```

Explanation

Returns the supplier names where the outstanding balance is greater than \$1000

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

```
SELECT m.materials_descr
FROM materials m
JOIN repair r ON m.material_ID = r.material_ID
WHERE r.number_hours > 40
```

Explanation

Returns all materials that were used in bids estimated to take longer than 40 hours

5. List all the material that Bob purchased from a specific supplier

```
SELECT m.materials_descr  
FROM materials m  
JOIN supplier s ON m.supplier_ID = s.supplier_ID  
WHERE s.supplier_name = 'Home Depot'
```

Explanation

Returns all materials purchased from a given supplier