Erik Smallwood Photography

Database Proposal

Project Phase II

Camperina

ID# XXXXXXXX

Email: XXXXXXXX

Table of Contents

[1. Introduction 3](#_Toc44624109)

[1.1 Problems 3](#_Toc44624110)

[1.2 Proposed Solution 3](#_Toc44624111)

[2. Database Design 3](#_Toc44624112)

[2.1 Data Dictionary 3](#_Toc44624113)

[2.2 Cross-Reference Table 5](#_Toc44624114)

[2.3 Entity Relationship Diagram (ERD) 6](#_Toc44624115)

[2.4 ERD Relational Model 7](#_Toc44624116)

[2.5 Functional Dependencies 8](#_Toc44624117)

[2.6 Constraints 8](#_Toc44624118)

[2.7 Relation Forms 9](#_Toc44624119)

[2.8 Normalized Relational Database 10](#_Toc44624120)

[References 11](#_Toc44624121)

# Introduction

Erik Smallwood Photography is a local small business that provides a variety of photography services. Currently, Erik, the owner, and his wife, Hannah, run the business and have a manual process for maintaining all aspects of their business.

## Problems

There is a manual process in place for setting appointments, preparing and finalizing contracts, freelance photographers, photo package options and selection, payments, and photograph storage. There are also many paper forms that are used to gather and document information about the services. Erik says, as the business grows, the current manual system is hindering, time and space consuming.

## Proposed Solution

The proposed solution is to create a database to assist the Smallwood’s with their business transactions. The database will house information about clients, jobs, and freelance photographers. It will also put all paper forms in an electronic format and have report options to help with time and information management.

# Database Design

## Data Dictionary

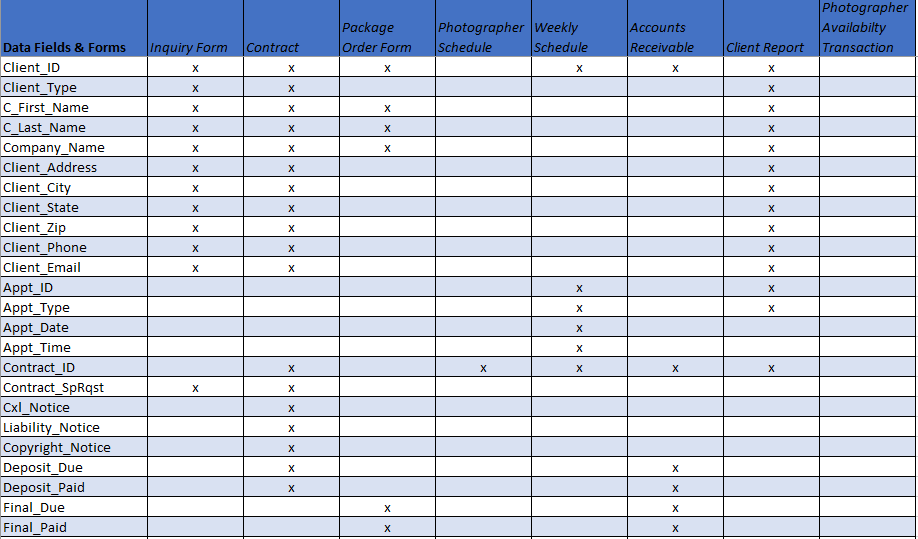
It is useful to know what type of information is held in a database and where it is stored. Below is the data dictionary to provide those details of the proposed database. It shows the name of each filed, what type and size of data it holds, and which table stores that information.

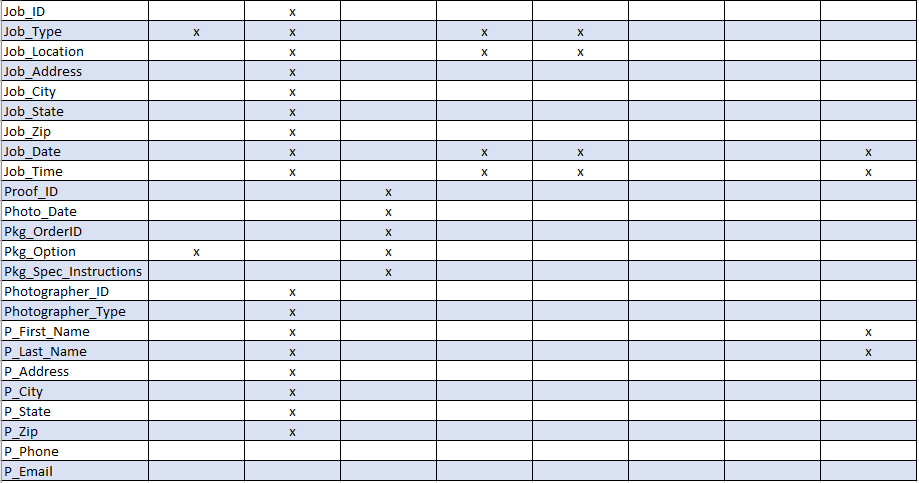
|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Size** | **Source** |
| Client\_ID | VarChar | 6 | Client |
| Client\_Type | Alpha | 10 | Client |
| C\_First\_Name | Alpha | 15 | Client |
| C\_Last\_Name | VarChar | 30 | Client |
| Company\_Name | VarChar | 50 | Client |
| Client\_Address | VarChar | 50 | Client |
| Client\_City | Alpha | 20 | Client |
| Client\_State | Alpha | 2 | Client |
| Client\_Zip | Numeric | 5 | Client |
| Client\_Phone | Numeric | 10 | Client |
| Client\_Email | VarChar | 50 | Client |
| Appt\_ID | Numeric | 6 | Appointment |
| Appt\_Type | Alpha | 10 | Appointment |
| Appt\_Date | Numeric | 8 | Appointment |
| Appt\_Time | Numeric | 4 | Appointment |
| Contract\_ID | VarChar | 10 | Contract |
| Contract\_SpRqst | VarChar | 100 | Contract |
| Cxl\_Notice | VarChar | 100 | Contract |
| Liability\_Notice | VarChar | 100 | Contract |
| Copyright\_Notice | VarChar | 100 | Contract |
| Deposit\_Due | Numeric | 6 | Contract |
| Deposit\_Paid | Numeric | 6 | Contract |
| Final\_Due | Numeric | 6 | Contract |
| Final\_Paid | Numeric | 6 | Contract |
| Job\_ID | Alpha | 6 | Job |
| Job\_Type | Alpha | 20 | Job |
| Job\_Location | VarChar | 30 | Job |
| Job\_Address | VarChar | 50 | Job |
| Job\_City | Alpha | 20 | Job |
| Job\_State | Alpha | 2 | Job |
| Job\_Zip | Numeric | 5 | Job |
| Job\_Date | Numeric | 8 | Job |
| Job\_Time | Numeric | 4 | Job |
| Proof\_ID | VarChar | 10 | Photo |
| Photo\_Date | Numeric | 8 | Photo |
| Pkg\_OrderID | VarChar | 8 | Package |
| Pkg\_Option | Alpha | 2 | Package |
| Pkg\_Spec\_Instructions | VarChar | 100 | Package |
| Photographer\_ID | VarChar | 10 | Photographer |
| Photographer\_Type | Alpha | 10 | Photographer |
| P\_First\_Name | Alpha | 15 | Photographer |
| P\_Last\_Name | Alpha | 30 | Photographer |
| P\_Address | VarChar | 50 | Photographer |
| P\_City | Alpha | 20 | Photographer |
| P\_State | Alpha | 2 | Photographer |
| P\_Zip | Numeric | 5 | Photographer |
| P\_Phone | Numeric | 10 | Photographer |
| P\_Email | VarChar | 50 | Photographer |

*Image 1 – Erik Smallwood Photography Data Dictionary*

## Cross-Reference Table

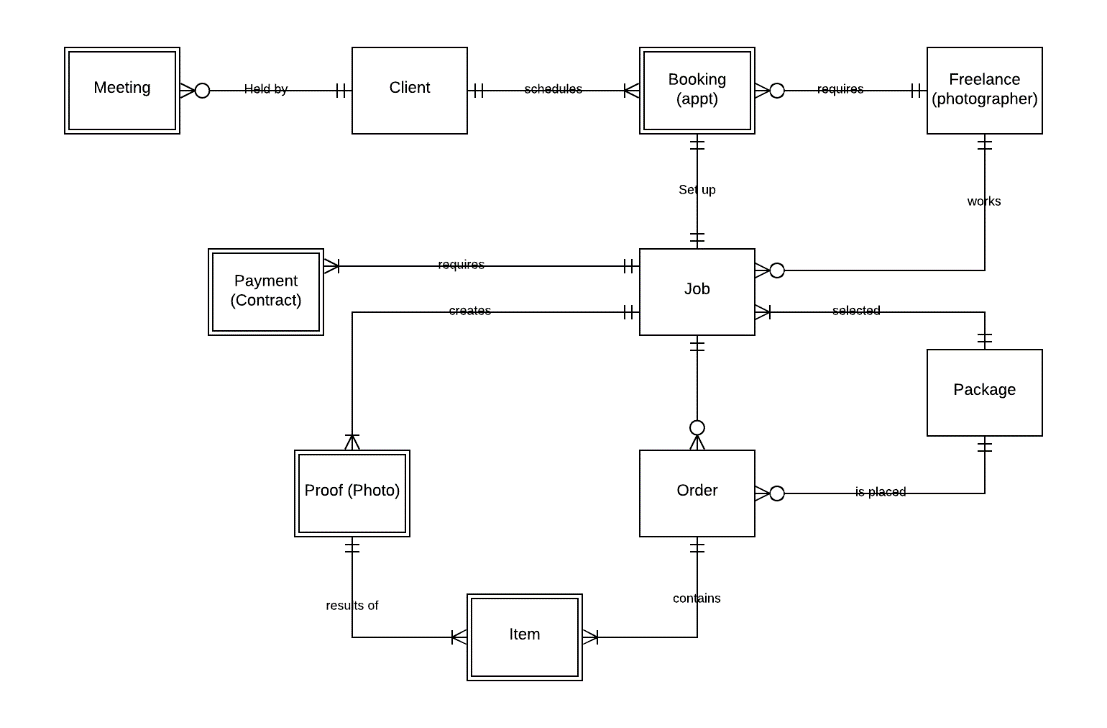
A cross-reference table is useful when you want to see how data from one area is related to another area. It shows how important it is to have accurate data entered because it can affect multiple areas of the database. Below is the cross-reference table for the proposed database between various data fields and the forms and reports that will be available.[1]



*Image 2 – Data Fields – Forms/Reports Cross-Reference Table*

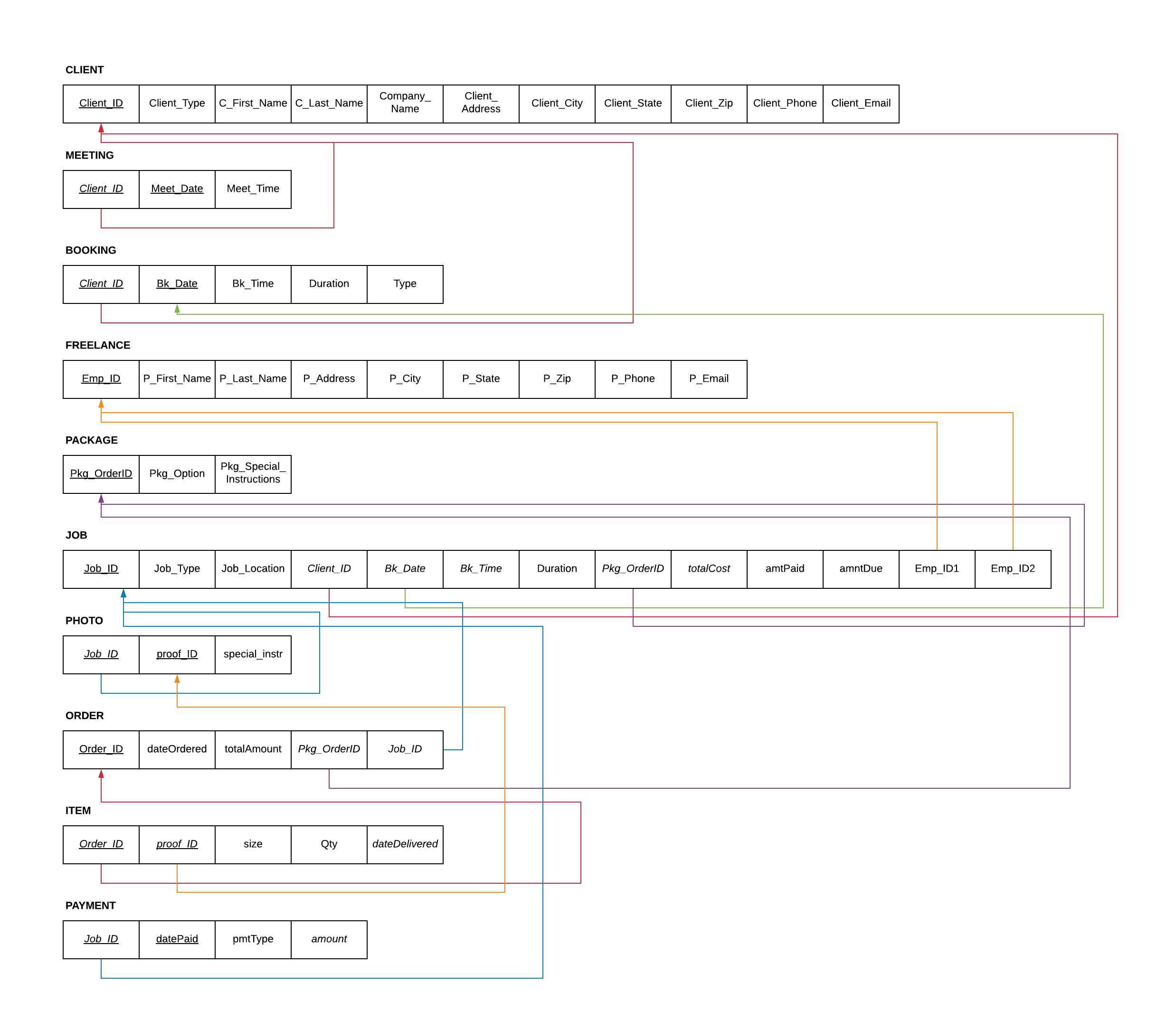
## Entity Relationship Diagram (ERD)

An entity relationship diagram is a visual tool that shows the layout of a database. The main areas of the proposed database and how they are related is shown below.



*Image 3 – Entity Relationship Diagram for Erik Smallwood Photography*

## ERD Relational Model

Another way to see how the entities of a database are related is by looking at a relational model. This model lists the entities, which would be tables, their fields and links them to other tables they are related to. The diagram blow is the ERD relation model for the proposed database.

*Image 4 – Entity Relation Model*

## Functional Dependencies

The following functional dependencies exist in the current database design:

Client 🡪 Meeting

Client 🡪 Booking

Client 🡪 Job

Booking 🡪 Job

Freelance 🡪Job

Package 🡪 Job

Package 🡪 Order

Job 🡪 Photo

Job 🡪 Order

Job 🡪 Payment

Photo 🡪 Item

Order 🡪 Item

## Constraints

In order to avoid constraints, it will be important to enter data in the proper sequence. The database is set up where each client will be assigned an ID. From there a meeting will be set-up and a booking scheduled. Once this has been completed, a freelance photographer will be selected, if needed, and the client will choose their package. The job will be completed which will create a job ID and photographs. After the photos are viewed and selected an order can be created for the items the client wants.

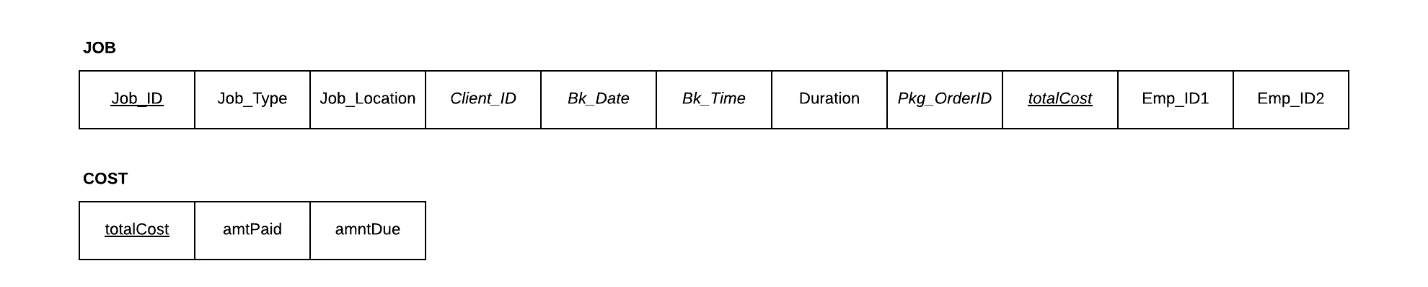
Entity integrity constraints could appear if areas of the database are accessed prior to all of the data being entered. An example of this would be a client comes in for a meeting and schedule a booking. The client would not be able to select a package option until a package order ID has been entered since that is the primary key needed for the package table in the database.

There could be referential integrity constraints if information is not entered in the proper order. For example, if a client selects an item prior to selecting the photo there would not be a proof ID to reference so that would cause a problem.

A semantic integrity constraint would occur if a photographer is booked for jobs that overlap. The photographer’s schedule should be referenced prior to assigning them to a job.

## Relation Forms

Of the ten relations in the database, nine of them are in third normal form (3NF) since no partial dependencies on a key exist. The only relation not in 3NF is Job. The values pertaining to cost could be broken out into a table of their own with totalCost being the primary key and amtPaid and amtDue being other fields of the ‘cost’ table.

The Job relation can be normalized into Boyce-Codd Normal Form (BCNF) by decomposing the relation as follows:

*Image 5 – Job Relation in BCNF*

## Normalized Relational Database

CLIENT (Client\_ID, Client\_Type, C\_First\_Name, C\_Last\_Name, Company\_Name, Client\_Address, Client\_City, Client\_State, Client\_Zip, Client\_Phone, Client\_Email)

MEETING (*Client\_ID*, Meet\_Date, Meet\_Time)

BOOKING (*Client\_ID*, Bk\_Date, Bk\_Time, Duration, Type)

FREELANCE (Emp\_ID, P\_First\_Name, P\_Last\_Name, P\_Address, P\_City, P\_State, P\_Zip, P\_Phone, P\_Email)

PACKAGE (Pkg\_OrderID, Pkg\_Option, Pkg\_Special\_Instructions)

JOB (Job\_ID, Type, Job\_Location, Bk\_Time, Duration, Emp\_ID1, Emp\_ID2, *Client\_ID, Bk\_Date, Pkg\_OrderID, totalCost*)

COST (totalCost, amtPaid, amtDue)

PHOTO (*Job\_ID*, proof\_ID, special\_instr)

ORDER (Order\_ID, dateOrdered, totalAmount, *Pkg\_OrderID, Job\_ID*)

ITEM (*Order\_ID*, *proof\_ID*, size, qty, dateDelivered)

PAYMENT (*Job\_ID*, datePaid, pmtType, amount)

# References

[1] K. Downs, *Database Skills: A Sane Approach to Choosing Primary Keys,* The Database Programmer, January 14, 2008. Accessed on: June 6, 2020. [Online]. Available: <http://database-programmer.blogspot.com/2008/01/database-skills-sane-approach-to.html#rule5>