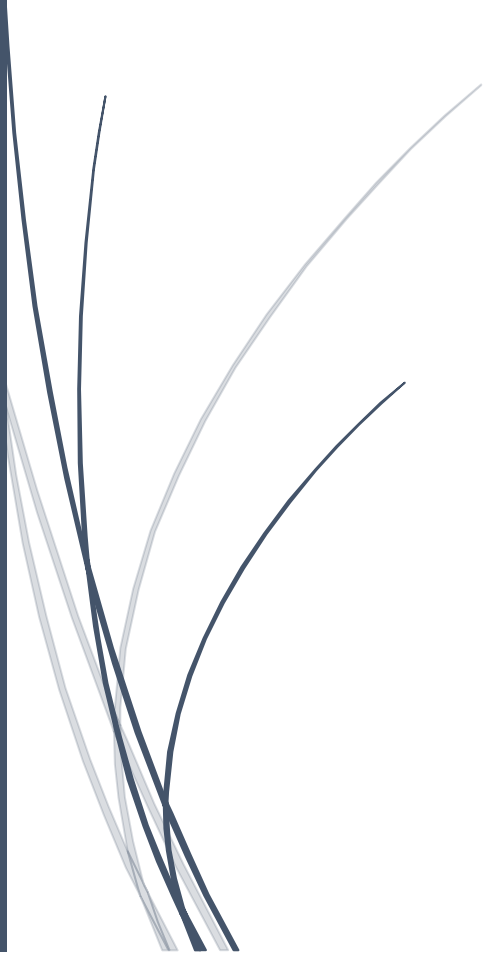
4/16/2018

Gemstone Construction

Database Documentation



Documentation by Geoff Groenendale
WSU VANCOUVER - MIS 372

Executive Summary

Gemstone Construction is a general contracting company that manages the construction projects of business clients. These projects range from remodels to new construction and everything in between. The company completes these projects with various scopes by contracting the requirements of the project out to other parties. These parties are called subcontractors as they do not have a direct agreement with the client but are bound to the client's requirements through the general contractor (Gemstone Construction). The purpose of this documentation is to capture the details of the SQL database for Gemstone Construction and explain how the database supports the project management activities of the company. This report documents the tables in the database, what the data types are for each field in those tables, and what operational reports are possible with the stored data. This documentation also describes the methods and decisions behind the design of the database.

Gemstone Construction utilizes a Microsoft SQL database to store information on clients, their projects, and the subcontractors involved in each of those projects. The database is designed to store the various relationships as it pertains to each project as well as information on project scope fulfillment. The database also serves as an address book and contact list for clients and subcontractors. This information is crucial for the account managers and the operations team. Firstly, the account managers need the list of clients for follow ups and new projects. Next the operations team needs a history of subcontractors to request bids for new projects from.

There are crucial processes that generate data for the database, most of which fall under the umbrella of project estimation. New projects are created and link clients to employees. Accurate creation of projects in the database is crucial as the scope elements create the subcontractor requirements. If this information is entered wrong then required subcontractors will not be notified, bids and contracts will not be generated, and projects will not be awarded to Gemstone by the client. After the right subcontractor types have offered bids, their details are added to the database and categorized using the Construction Specification Institute's numbering and classification system. This system uses 5-digit numbers where the first two numbers are a category of scope and the last three numbers specify a common construction activity such as pouring foundation or installing HVAC systems. Finally, summary reports on required project scope elements are generated in order to find what project elements need to be fulfilled as well as a complete final bid report for the client.

The database was created with care to ensure that bids from subcontractors are matched to a specific project and that each of those bids are individually identified. Firstly, there is a unique number for each project that starts with a three-digit number that matches the id of the signed client. Next, there is a two-digit number for the year in which the project started. The start of the project is defined as the moment when an account manager and the executive team decides it is a good project for the company and the operations team is notified to get bids from subcontractors. The last set of digits matches the unique project ID in order to ensure that each project number is unique.

Table of Contents

Change Log

Narrative

- Organization restructured for B2B projects
- Organization overview updated to match restructuring
- Process overview changed
    - Removed mention of materials
    - Added subcontractor bid process
    - Modified bid package to client process
- Transactions and objects updated from restructure
    - Removed Materials and Project Materials
    - Added Objects – Subcontractors and Employees
    - Added Transactions – Subcontractor Bids, Bid Details, Project Scopes
    - Updated Clients and Client Projects

Flow Chart

- Updated to reflect changes in organization
    - Removed materials datastore
    - Added subcontractor swim lane
    - Added document flows and symbols
    - Added subcontractors, employees, and project bids datastores

ER Diagram

- Updated to reflect changes in organization
    - Removed entities – materials and project materials
    - Added entities – project bids, subcontractors, and employees
- Updated attributes to reflect changes
- Added attribute datatypes
- Formatted identity and derived attributes
- Updated design for readability

Database Diagram

- Diagram updated to show database modifications
- Relationship connectors show connections between fields

Data Design Explanation

- Updated object tables to reflect changes in organization
    - Removed account managers

    - Modified client_projects table to include a project number that is another unique identifier
- Updated transaction tables to reflect changes in organization
    -
- Updated intersection tables to reflect changes in organization
    - Removed client projects
    - Modified account managers to project roles table
    - Added bid details, project scopes,
- Updated category tables to reflect changes in organization
    - Modified subcontractor trades to Construction Specification Institute (CSI) codes
    - Added CSI categories to further categorize CSI codes

Report Documentation

- Updated example report format for readability and

Organizational Overview

Gemstone Construction is a general contractor specializing in construction project management for business clients. These projects range from remodels to new construction and everything in between. A general contractors primary role in a project is to manage all of the details, schedules, and work within a construction project. The general contractor may do some of the physical work on the project. Most of the time it is advantageous to hire other contractors to do part of the work. This reduces the equipment needs and licensing requirements of the general contractor. Gemstone construction This means that the general contractor may do most of the framing work but will hire an electrician and plumber to do some of the specific tasks. Gemstone construction handles a variety of project types. Remodels can be simple two-week projects with minimal requirements. Ground-up new construction may take as long as a year. Contract values range from $10,000 to $5,000,000. Gemstone charges a contractor's fee of 8% to 12% of total contract amount.

Process Overview

Before the company starts physical work on a project the company must win the approval of a client. A project begins after the client has spoken with designers and architects and most of the project design process has already been completed. Clients contact various general contractors in search of the best fit for their project. The client is at a stage in the building process where they are gathering quotes from various companies to find the best option. Depending on the client this can mean the lowest price, most detailed bids, or fastest schedule.

The sales department will reach out to as many clients as possible. As a connection is made with each client they are added to a datastore along with their contact information. The sales department will work with each client to offer the companies services for building out the home. Once a client asks for a quote by providing building plans and design specifications, the account manager and operations executive will decide if the project could be valuable to the company. An entry is made in the client projects database when the operations department begins collecting project bids. Clients can have several projects with the company. Therefore, a unique identifier is required to differentiate projects of a client, between clients, and between operational years. Upon receipt of the plans the operations department will review the plans to determine what activities are required for the project. From this a list of scope details is created that includes the type of activity for the project. This also requires a unique identifier which is facilitated by the Construction Specification Institutes categorization system. These are five-digit numbers where the first two digits describes a category of scope and the last three digits breakdown the category even more. For example, the value 03200 specifies concrete reinforcement for the concrete category. Scope lists are then sent out to a network of contractors that can handle specific scopes. If any contractor sends a bid for part of the work, the bid is added to the database and matches a subcontractor to a client project and categorizes the bid details with the CSI codes. Once the project has all of the bids required to fulfill the scope elements a bid package is sent to the client for review and approval/denial. The actual physical work of the project can begin when the client has approved the bid package.

Transactions and Objects

Objects

- Clients
    - Client ID – integer, unique identifier
    - Client Name – nvarchar(50)
    - Address
        - Street1 – nvarchar(50)
        - Street2 – nvarchar(50)
        - City – nvarchar(50)
        - State – nchar(2)
        - Zip – nvarchar(10)
    - Phone – nvarchar(20)
    - Email – nvarchar(50)
    - Project count – integer
- Employees
    - Employee ID – integer, unique identifier
    - First Name – nvarchar(50)
    - Last Name – nvarchar(50)
    - Email – nvarchar(50)
    - Phone – nvarchar(20)
    - Emp_cat_id – integer
    - Project count – integer
- Subcontractors
    - Subcontractor ID – integer, unique identifier
    - Name – nvarchar(50)
    - Address
        - Street1 – nvarchar(50)
        - Street2 – nvarchar(50)
        - City – nvarchar(50)
        - State – nchar(2)
        - Zip – nvarchar(10)
    - Phone – nvarchar(20)
    - Email – nvarchar(50)
    - Project count – integer

Transactions

- Client Projects
    - Project ID – integer, unique identifier
    - Project Number – computed column (integer-year-integer)
    - Project Name – nvarchar(50)
    - Project Address
        - Street1 – nvarchar(50)
        - Street2 – nvarchar(50)
        - City – nvarchar(50)
        - State – nchar(2)

        - Zip – nvarchar(10)
    - Client ID – integer
    - Start Date – datetime
    - End Date – datetime
    - Sq footage – decimal(10,2)
    - Cost Total – money
    - Contract Amount – money
- Project Bids
    - Bid ID – integer, unique identifier
    - Subcontractor ID – integer, unique identifier
    - Project ID – integer, unique identifier
    - Bid Amount – money
    - Bid approved – bit
- Bid details
    - Detail ID – integer, unique identifier
    - CSI Code – varchar(5)
    - Bid ID – integer, unique identifier
    - Qty – integer
    - Qty breakdown – nvarchar(50)
    - Detail amount – money

Estimation Process Flow Chart

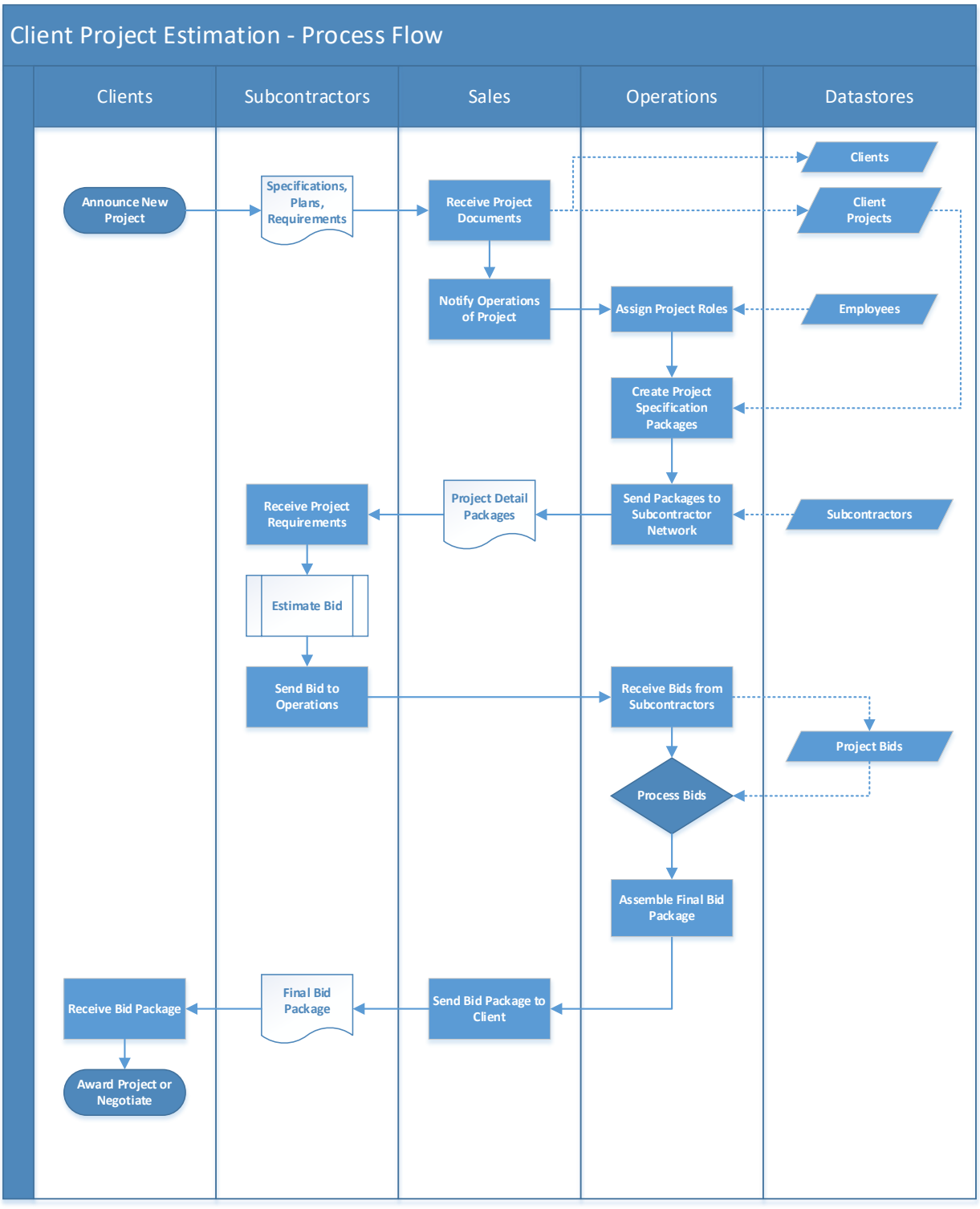


Figure 1: Swimlane Flowchart depicting the process of estimating new client projects based on materials required. Sales and Operations personnel are required for this process. Datastores detailed in the far right column.

ER Diagram

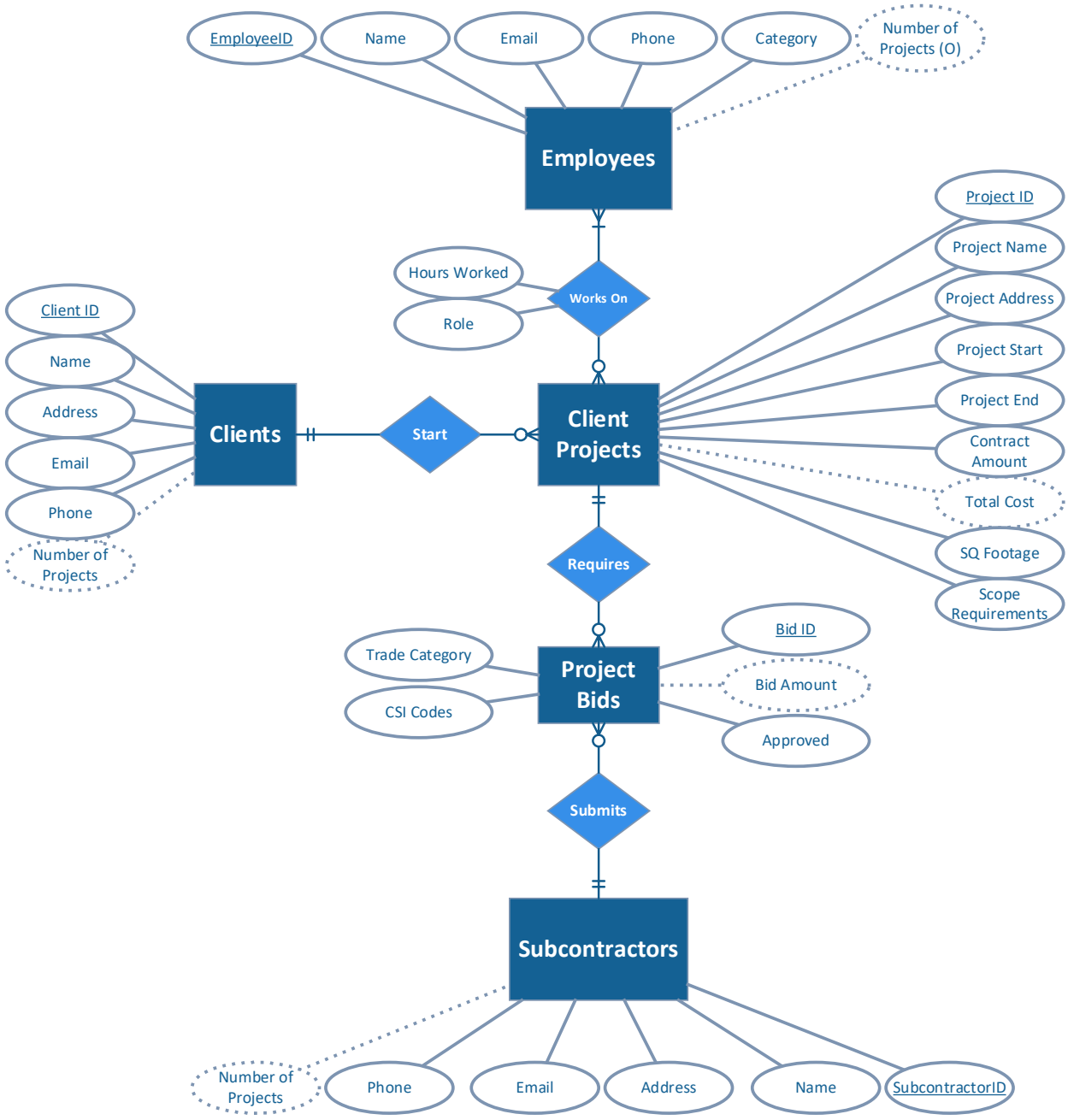


Figure 2: ER diagram including all entities and attributes required for Estimating process. Note the entities match the datastores in the above diagram.

Database Diagram

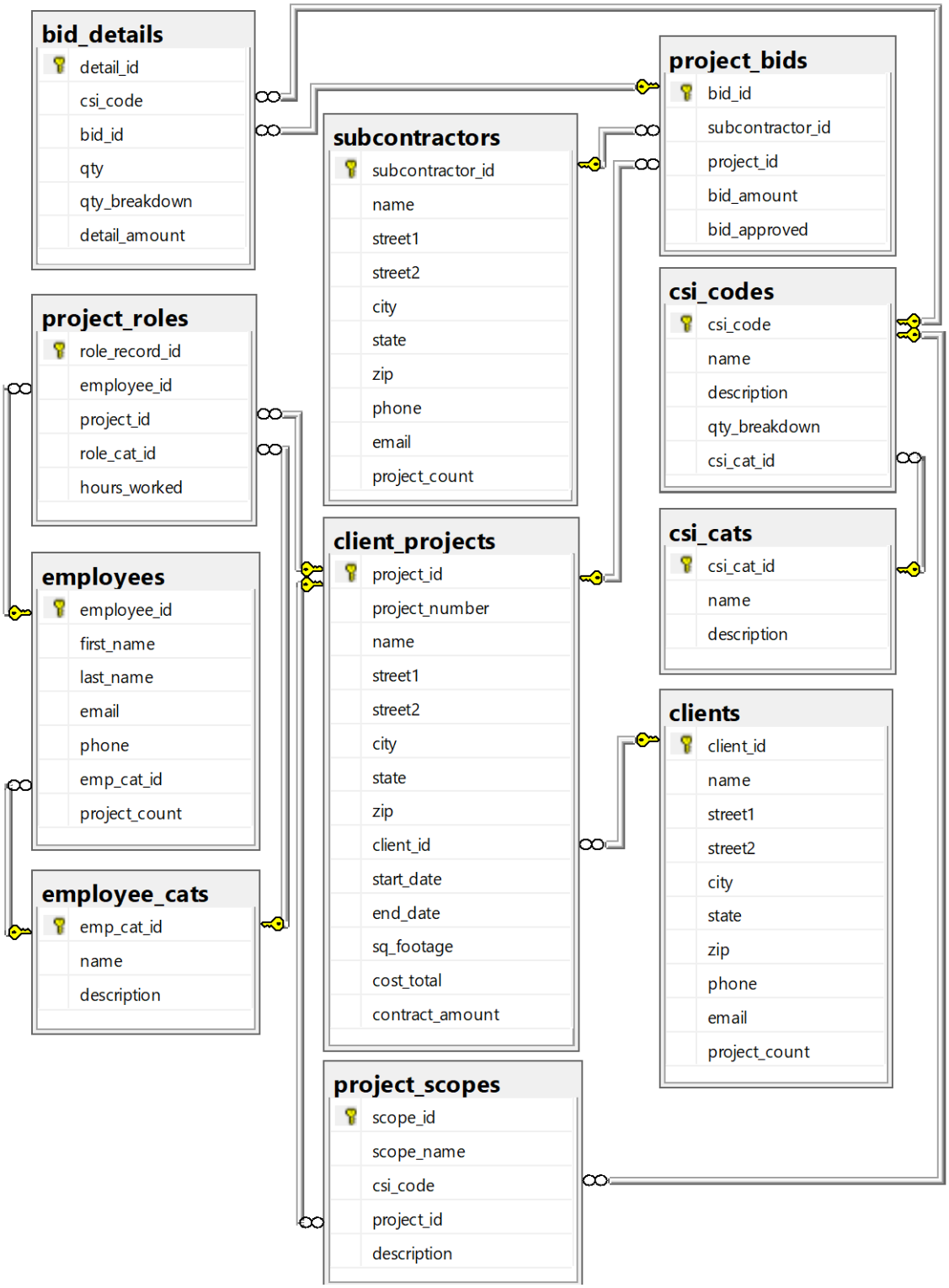


Figure 3: This is a Caption

Design Explanation

Object/Transaction tables

This database keeps track of three objects: subcontractors, clients, and employees. There are two transaction tables in the database: client projects, and project bids.

Object Tables

- **Clients** – This table stores contact info for all clients. There is also a project count to record the history of each client.
- **Employees** – This table stores all the employees with their names and contact info. The table also has employee categories recorded in order to record their role in the company. There is also a project count to record the history of each employee.
- **Subcontractors** – This table stores all of the subcontractor businesses with contact information and includes a value to differentiate which trade in a project they handle. There is also a project count to record the history of each subcontractor.

Transaction Tables

- **Client Projects** – Client Projects is a transaction table that records every house project that a client wants to build. The table records start and stop dates, the address, the total price, square foot.
- **Project Bids** – Subcontractors of different trades will provide bids on a project depending on the scope of the project.

Intersection Tables

The design features three intersection tables: Project Bids, Bid Details, Project Scopes, and Project Roles.

- The **project bids** table intersects between the subcontractors table and the client projects table. This allows any subcontractor to bid on more than one project, and any project can have any number of subcontractors attached to it.
- The **bid details** table further describes the bid from a subcontractor by detailing what CSI codes and scopes the bid covers. One bid may cover several scopes from a subcontractor as in a concrete subcontractor cutting concrete and pouring a new slab. There can also be several bids that include framing scopes.
- The **project scopes** table intersects between the client projects and CSI codes tables. This table serves the M:N relationship between projects and CSI codes. Since many projects may need concrete foundation repairs and one project has several scope requirements this table serves the role of matching this information together.
- The **project roles** table intersects between the employees table and the client projects table. This facilitates a M:N relationship because there are many employees that might be involved in a client project and there are many projects that an employee might be attached to. This table allows the database to track which account managers, project managers, project coordinators, and superintendents are involved in which projects. This also facilitates the project count calculated value within the employees table.

Category Tables

The database features two category tables: CSI Categories, CSI Codes, and employee categories.

- **CSI Codes** allows project bids and project scopes to be categorized by five number values that identify specific activities within a project. The first two numbers are also related to the CSI category while the last three numbers further breakdown the category into specific activities. CSI codes come from the Construction Specification Institute and are one standard used by the construction industry.
- **CSI Categories** is the overarching structure to the CSI codes table. These categories are identified by the first two numbers in the CSI code. CSI codes come from the Construction Specification Institute and are one standard used by the construction industry.
- The **employee categories** table is important for HR and the other functions of the company. The table categorizes different employees by job category. These categories are also crucial for identifying the role that an employee served for a project. An employee may act as a project coordinator on one project and later in their employment history may act as a project manager on a different project.

Report Documentation

The bid to client report is a fundamental report for the sales team. The report collects all of the project bids for a client project from the database. The report only shows those bids that were approved as the operation team may approve one bid that is more affordable or comes from a reliable subcontractor. These bids are then sorted by the CSI scope category and then CSI codes. The report then displays the name and phone of each approved subcontractor, amount of each bid detail, and the total bid from each subcontractor. The heading for the report displays the contact information for the client and the name of the Account Manager as well as identifying information for the project like project number, name, and address. This report is crucial for the account manager as these reports show which projects have enough bids for the project and which projects are waiting on more bids. The report is also important for reporting to the client what the total for the project is. The report depends on what scope the client project has in it as well as the bid price that each subcontractor sends in for the project.

Example Report

Project Number: 103-18-8		Bid To Client		
Client Name: Pacific Development Client Phone: (360) 524-7899		Account Manager: Andrew Landdog		
Project Name: Dicks Hardware Project Address: 540 Jefferson Ave Portland, Or 97217		Project Total: 1900000 Project SqFt: 5000		
Subcontractor Name	Subcontractor Trade Name	Bid Amount	Subcontractor Phone	Subcontractor Email
Freds Concrete	concrete	30000	5038997526	fred@gmail.com
Portland Framers	framers	19000	9712568999	info@portlandframers.com
Bob Ross	painter	30000	5038994566	trees@bobspainting.com
American Plumbing	plumber	14000	5032896498	info@americanplumbing.com

Figure 4: Bid to Client Report - This report details which subcontractor bids have been received and approved. The bids are reported by trade alphabetically. Other companies may choose to use the Construction Specifications Institute numbering

Report Header: “Bid to Client - Header” query

```
SELECT dbo.client_projects.project_number AS [Project Number],
         dbo.clients.name AS [Client Name],
         ('(' + LEFT(dbo.clients.phone,3) +') ' + SUBSTRING(dbo.clients.phone, 4,3) +
           '-' + RIGHT(dbo.clients.phone,4)) AS [Client Phone],
         (dbo.employees.first_name + ' ' + dbo.employees.last_name)
           AS [Account Manager]
         dbo.client_projects.name AS [Project Name],
         (dbo.client_projects.street1 + ISNULL((',' + dbo.client_projects.street2 +
           ','), '') + ', ' + dbo.client_projects.city + ' ' +
           dbo.client_projects.state + ' ' + dbo.client_projects.zip)
           AS [Project Address]
         dbo.client_projects.contract_amount AS [Project Total],
         dbo.client_projects.sq_footage AS [Project SqFt]
FROM   dbo.project_roles INNER JOIN
           dbo.client_projects ON dbo.client_projects.project_id =
           dbo.project_roles.project_id INNER JOIN dbo.employees ON
           dbo.project_roles.employee_id = dbo.employees.employee_id INNER JOIN
           dbo.employee_cats ON dbo.employees.emp_cat_id = dbo.employee_cats.emp_cat_id
           INNER JOIN dbo.clients ON dbo.client_projects.client_id =
           dbo.clients.client_id
WHERE  (dbo.client_projects.project_id = 8) AND (dbo.project_roles.role_cat_id = 1)
```

SQL Query 1: "Bid to Client - Header" Note the concatenation functiuons and conditionals used to combine various data into one column to make output more concise.

Table 1 – Returned Data from header query

Project Number	Client Name	Client Phone	Account Manager	Project Name	Project Address	Project Total	Project SqFt
103-18-8	Pacific Development	(360) 524-7899	Andrew Landdog	Dicks Hardware	888 Easy St., Portland, Or 97210	NULL	NULL

Table 1: Note the header query for this report returns a lot of data. Note the use of aliases and concatenation to combine the values into one column of the result.

Report Body: “Bid to Client - Body” query

```
SELECT TOP (100) PERCENT
        CONVERT(nvarchar, dbo.project_bids.bid_id) + '-' +
          CONVERT(nvarchar, dbo.bid_details.detail_id) AS [Bid Detail ID],
        dbo.csi_cats.name AS [CSI Category],
        dbo.csi_codes.csi_code AS [CSI Code],
        dbo.subcontractors.name AS [Subcontractor Name],
          '(' + LEFT(dbo.subcontractors.phone, 3) + ') ' +
          SUBSTRING(dbo.subcontractors.phone, 4, 3) + '-' +
          RIGHT(dbo.subcontractors.phone, 4) AS [Sub Phone],
        dbo.subcontractors.email, dbo.bid_details.detail_amount AS [Bid Amount]
FROM  dbo.client_projects INNER JOIN
        dbo.project_bids ON dbo.client_projects.project_id =
        dbo.project_bids.project_id INNER JOIN dbo.subcontractors ON
        dbo.project_bids.subcontractor_id = dbo.subcontractors.subcontractor_id INNER
        JOIN dbo.bid_details ON dbo.project_bids.bid_id = dbo.bid_details.bid_id INNER
        JOIN dbo.csi_codes ON dbo.bid_details.csi_code = dbo.csi_codes.csi_code INNER
        JOIN dbo.csi_cats ON dbo.csi_codes.csi_cat_id = dbo.csi_cats.csi_cat_id
WHERE (dbo.client_projects.project_id = 8) AND (dbo.project_bids.bid_approved = 1)
ORDER BY [CSI Category], [CSI Code]
```

SQL Query 2: "Bid to Client - Body" Note the use of the WHERE clause to specify the project with the ID of 8 and only the bids that are approved.

Table 2 – Returned Data from body query

Bid Detail ID	CSI Category	CSI Code	Subcontractor Name	Sub Phone	email	Bid Amount
10037-10	Concrete	03200	Rocks Concrete	(971) 555-3456	dwayne@rockconcrete.com	28000.00
10024-8	Wood plastics	06100	Portland Framers	(971) 256-8999	info@portlandframers.com	19000.00

Table 2: Data returned from body query. Note that each subcontractor bid shown is an approved bid. If a bid is not approved, it isn't part of the report or a part of the project.

SQL Aggregators

Aggregator 1 – Calculate Bid Totals

This aggregator query calculates the bid totals from a subcontractor for a project based on the individual details of each bid.

```
SELECT TOP (100) PERCENT
       dbo.project_bids.bid_id,
       SUM(dbo.bid_details.detail_amount) AS [Bid Total],
       COUNT(dbo.bid_details.detail_ID) AS [Number of Details]
       FROM dbo.bid_details INNER JOIN
       dbo.project_bids ON dbo.bid_details.bid_id = dbo.project_bids.bid_id
WHERE dbo.bid_details.bid_id = dbo.project_bids.bid_id
GROUP BY dbo.project_bids.bid_id
```

SQL Query 3: This is the SQL Code copied from the Bid Total aggregator query. Note the query should total the total dollars and report the number of details in the bid.

Bid ID	Bid Total	Number of Details
10016	20000.00	1
10018	20000.00	1
10020	20000.00	1
10022	19000.00	1
10024	19000.00	1
10031	31000.00	1
10037	28000.00	1

Table 3: Example data generated by the SQL aggregator. Note that these results show one bid detail for those bids that had lump sum bids.

Aggregator 2 – Sum Employee Project Count

This Aggregator totals the number of projects that an employee has been involved with. A similar query can be generated for all subcontractors and clients. This is a useful business intelligence tool to understand which clients, subcontractors, and employees have been the most valuable.

```
SELECT TOP (100) PERCENT (dbo.employees.first_name + ' ' + dbo.employees.last_name) AS
[Employee Name],
       dbo.employee_cats.name AS [Role in Company],
       COUNT(dbo.project_roles.role_record_id) AS [Number of Projects]
       FROM dbo.employees INNER JOIN
       dbo.project_roles ON dbo.employees.employee_id = dbo.project_roles.employee_id
INNER JOIN
       dbo.client_projects ON dbo.project_roles.project_id =
dbo.client_projects.project_id INNER JOIN
       dbo.employee_cats ON dbo.employees.emp_cat_id = dbo.employee_cats.emp_cat_id
GROUP BY dbo.employees.first_name, dbo.employees.last_name, dbo.employee_cats.name
ORDER BY [Employee Name]
```

SQL Query 4: This is the SQL Code copied from the Sum Employee Project Count aggregator query. Note that this query could be modified to total the project totals for clients and subcontractors as well. Also note this query can also show involvement for each employee by their roles in projects. This is valuable to see an employees progress as they move up in the company.

Employee Name	Role in Company	Number of Projects
Andrew Landdog	Account Manager	2
Fred Ferndale	Account Manager	1
Misty May	Account Manager	1
Ralph Douglas	Superintendent	4

Table 4: Example data generated by the SQL aggregator. Note that Ralph is the superintendent for several projects and these results may show the need for more superintendents.

SQL Stored Procedures

Stored Procedure 1 - Add Subcontractor Bid

This first stored procedure features 3 sub procedures in order to separate the operations and ensure that each operation happens in succession. The benefit of separating this procedures into sub procedures also ensures future customization and the possibility of bulk entries utilizing the sub procedures. The First query shown is the parent stored procedure followed by the three sub procedures displayed respectively to the first.

```
USE [geoffgroenendale]
GO
/****** Object:  StoredProcedure [dbo].[addSubcontractorBid]    Script Date: 4/29/2018
8:57:29 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:           Geoff Groenendale
-- Create date: 4/29/2018
-- Description:      The purpose of this procedure is to add a bid from a
-- subcontractor when supplied with the subcontractor ID, ProjectID, CSI Code,
-- quantity and quantity breakdown if there are any, and finally the dollar amount.
-- Many entries are simply lump sums but this procedure also allows detailed
-- subcontractor bids to be recorded.
-- =============================================
ALTER PROCEDURE [dbo].[addSubcontractorBid]
       -- Add the parameters for the stored procedure here
(
       @SubcontractorID int,
       @ProjectID int,
       @CSICode nvarchar(10),
       @qty decimal(4,2),
       @qtybrkdown nvarchar(20),
       @BidDollars money
)
AS
BEGIN
       -- SET NOCOUNT ON added to prevent extra result sets from
       -- interfering with SELECT statements.
       SET NOCOUNT ON;
       --First execute a procedure that creates a bid relationship between
```

```
        --subcontractor and the project then return the unique Bid ID
        DECLARE @BidID INT
    EXECUTE dbo.addProjectBid @SubcontractorID, @ProjectID, @BidID Output
        --Next execute a procedure that creates a bid detail from the information
        --provided and the newly generated Bid ID. Return the unique detailID
        DECLARE @DetailID INT
        EXECUTE dbo.addBidDetails @CSICode, @qty, @qtybrkdown, @BidDollars, @BidID,
@DetailID Output
        --Finally Update the Bid Amount in the Project Bids table to reflect the
        --total of all the details for the bid.
        DECLARE @BidTotal Money
        EXECUTE dbo.updateProjectBidTotal @BidID, @DetailID, @BidDollars, @BidTotal
Output
END
```

Sub Procedures

Add Project Bid

```
USE [geoffgroenendale]
GO
/****** Object:  StoredProcedure [dbo].[addProjectBid]    Script Date: 4/29/2018
9:10:22 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:           Geoff Groenendale
-- Create date: 4/29/2018
-- Description:      This procedure creates the relationship between
-- the subcontractor and the project as a bid. It returns an ID.
-- =============================================
ALTER PROCEDURE [dbo].[addProjectBid]
        -- Add the parameters for the stored procedure here
(
        @subcontractorid INT,
        @projectid INT,
        @bidid INT OUTPUT
)
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        SET NOCOUNT ON;

    -- Insert statements for procedure here
        INSERT INTO dbo.project_bids (subcontractor_id, [project_id])
        VALUES (@subcontractorid, @projectid)
        SET @bidid = SCOPE_IDENTITY();
END
```

Add Bid Details

```
USE [geoffgroenendale]
GO
/****** Object:  StoredProcedure [dbo].[addBidDetails]    Script Date: 4/29/2018
9:12:17 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:           4/29/2018
-- Description:      This procedure adds details to the project bid.
-- It also matches the various CSI codes to the project bid
-- and facilitates that relationship. It then returns an id
-- =============================================
ALTER PROCEDURE [dbo].[addBidDetails]
       -- Add the parameters for the stored procedure here
(
       @CSICode nvarchar(10),
       @qty decimal(4,2),
       @qtybrkdown nvarchar(20),
       @BidDollars money,
       @BidID INT,
       @DetailID INT Output
)
AS
BEGIN
       -- SET NOCOUNT ON added to prevent extra result sets from
       -- interfering with SELECT statements.
       SET NOCOUNT ON;

       --The insert for the bid details occurs here.
    INSERT INTO dbo.bid_details (csi_code, bid_id, qty, qty_breakdown, detail_amount)
       VALUES (@CSICode, @BidID, @qty, @qtybrkdown, @BidDollars)
       SET @Detailid = SCOPE_IDENTITY();
END
```

Update Project Bid Total

```
USE [geoffgroenendale]
GO
/****** Object:  StoredProcedure [dbo].[updateProjectBidTotal]    Script Date:
4/29/2018 9:14:29 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:           Geoff Groenendale
-- Create date: 4/29/2018
-- Description:      The purpose of this procedure is to update
-- the project bid with a dollar amount once the bid details
-- have been added. Note this procedure seems to not work
-- or the update to the project bid table needs time to happen.
-- More research and development is required.
```

```
-- =============================================
ALTER PROCEDURE [dbo].[updateProjectBidTotal]
	-- Add the parameters for the stored procedure here
(
	@bidid int,
	@detailid int,
	@detailbid money,
	@BidTotal money OUTPUT
)
AS
BEGIN
	-- SET NOCOUNT ON added to prevent extra result sets from
	-- interfering with SELECT statements.
	SET NOCOUNT ON;
	--The procedure checks to make sure the bid actually exists before updating
	IF EXISTS (SELECT 1 FROM dbo.project_bids WHERE dbo.project_bids.bid_id =
@bidid)
		BEGIN
			UPDATE dbo.project_bids
			SET bid_amount = bid_amount + @detailbid
		END
END
```

Stored Procedure 2 - Report Project Details

The purpose of this stored procedure is to display what the scope requirements are for a project and then detail how many bids have been gathered for each of the projects.

```
USE [geoffgroenendale]
GO
/****** Object:  StoredProcedure [dbo].[ReportProjectDetails]    Script Date: 4/29/2018
9:19:35 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:           <Author,,Name>
-- Create date: <Create Date,,>
-- Description:      <Description,,>
-- =============================================
ALTER PROCEDURE [dbo].[ReportProjectDetails]
(
       -- Add the parameters for the stored procedure here
       @ProjectID INT
)
AS
BEGIN
       -- SET NOCOUNT ON added to prevent extra result sets from
       -- interfering with SELECT statements.
       SET NOCOUNT ON;
       --Declare Variables to contian various values from each table
       DECLARE @req_id INT
       DECLARE @req_name nvarchar(50)
       DECLARE @req_code nvarchar(50)
       DECLARE @bid_id int
       DECLARE @detail_id int
       DECLARE @detail_amt money
       --First find all of the required scopes for the project and generate a list.
       SELECT
                dbo.project_scopes.scope_id AS [Required Scope ID],
                dbo.project_scopes.scope_name AS [Required Scope],
                dbo.project_scopes.csi_code AS [Required CSI CODE]
       FROM dbo.project_scopes WHERE dbo.project_scopes.project_id = @ProjectID
       ORDER BY dbo.project_scopes.csi_code
       --Next find the individual project bids and their details that match that
required
       --scope then generate a list. Also list which bids are approved or not.
       SELECT
                dbo.bid_details.csi_code AS [CSI CODE],
                dbo.csi_codes.name AS [Fulfilled Requirement],
                (CONVERT(nvarchar,dbo.project_bids.bid_id) + '-' +
CONVERT(nvarchar,dbo.bid_details.detail_id)) AS [Bid-Detail ID],
                dbo.bid_details.detail_amount AS [Bid Detail Price],
                --This case makes it so that the bid approved bit value
                --is converted into an easily understood word
                CASE
                          WHEN dbo.project_bids.bid_approved = 1 THEN 'APPROVED'
                          WHEN dbo.project_bids.bid_approved = 0 THEN 'NOT APPROVED'
                          ELSE 'UNDEFINED'
                END AS [Approved (True/False)]
                FROM dbo.bid_details INNER JOIN
```

```
                dbo.project_bids ON dbo.bid_details.bid_id = dbo.project_bids.bid_id
INNER JOIN
                dbo.csi_codes ON dbo.bid_details.csi_code = dbo.csi_codes.csi_code
      WHERE (dbo.project_bids.project_id = @ProjectID)
      ORDER BY [CSI CODE]
END
```

Results | Messages

	Required Scope ID	Required Scope	Required CSI CODE
1	1	Concrete Foundation	03200
2	3	Concrete Foundation	03300
3	4	Framing	06100

	CSI CODE	Fulfilled Requirement	Bid-Detail ID	Bid Detail Price	Approved (True/False)
1	03200	Concrete Reinforcement	10037-10	28000.00	APPROVED
2	06100	Rough Carpentry	10016-4	20000.00	NOT APPROVED
3	06100	Rough Carpentry	10018-5	20000.00	NOT APPROVED
4	06100	Rough Carpentry	10020-6	20000.00	NOT APPROVED
5	06100	Rough Carpentry	10022-7	19000.00	NOT APPROVED
6	06100	Rough Carpentry	10024-8	19000.00	APPROVED

Figure 5: The above figure details all of the required scopes and all of the current bids for project 8.

Example Database Data

Note: The data shown in this section was pulled using a simple SELECT query in order to pull field names easily. The SQL command format is as follows: SELECT * FROM [table name]

Bid Details

detail_id	csi_code	bid_id	qty	qty_breakdown	detail_amount
4	06100	10016	1	lumpsum	20000.00
5	06100	10018	1	lumpsum	20000.00
6	06100	10020	1	lumpsum	20000.00
7	06100	10022	1	lumpsum	19000.00
8	06100	10024	1	lumpsum	19000.00
9	06100	10031	1	lsum	31000.00
10	03200	10037	1	lsum	28000.00

Table 5: - Account Managers example data. Note: this table serves as an intersection table between employees and client projects for Account Managers. It also facilitates the use of the YTD sales column that is used to track the managers commission.

Clients

client_id	101	102	103	104
name	Clark Properties	Gin & Euler Co	Pacific Development	NorthWest Properties
street1	5664 Young St	875 Killer Ave	5420 Splash Dr	5050 123rd Ave.
street2	NULL	NULL	Ste. 100	NULL
city	Portland	Gresham	Gresham	Portland
state	OR	OR	OR	OR
zip	97217	97230	97230	97217
phone	5097894566	5038794555	3605247899	5037774589
email	info@clarkprops.net	james.gin@gmail.com	jerry@pacdev.com	fred@gmail.com
project_count	NULL	NULL	NULL	NULL

Table 7: Clients example data. Note: Bill Smith is the client shown above in the example report.

Client Projects

project_id	2	3	4	6	8
project_number	101-18-2	101-18-3	101-18-4	102-18-6	103-18-8
name	Bobs Burgers - Yamhill	Bobs Burgers - Broadway	Bobs Burgers - Mill Plain East	Extreme Supply - Portland	Dicks Hardware
street1	540 Jefferson Ave	8400 Broadway St	12500 SE Mill Plain Blvd	990 Washington St	888 Easy St.
street2	NULL	NULL	NULL	NULL	NULL
city	Portland	Portland	Vancouver	Portland	Portland
state	Or	Or	WA	Or	Or
zip	97217	97217	98684	97216	97210
client_id	101	101	101	102	103
start_date	NULL	NULL	NULL	NULL	NULL
end_date	NULL	NULL	NULL	NULL	NULL
sq_footage	NULL	NULL	NULL	NULL	NULL
cost_total	NULL	NULL	NULL	NULL	NULL

Table 6: Client Projects example data. Note that the figures for this table were transposed so as to fit on the document and make better sense of the data. Each project is a column in this table.

Employee Categories

emp_cat_id	name	description
1	Account Manager	The Account Managers handle all of the clients and communicate requirements back to operations.
2	Payroll Clerk	The employee that handles all of the payroll for the office.
3	Executive	The executive team includes all of the chief officers of the board.
4	Project Manager	The project managers handle all of the home builds after the client accepts the bid.
5	Superintendent	stays onsite at the project to make sure everything goes as planned.
10	Project Coordinator	Works with the Project Manager to make sure that a project has the resources it needs.

Table 7: Employee Categories example data. This is a category table for all employee types.

CSI Categories

csi_cat_id	name	description
02000	Site Work	Includes scope items such as site prep, earthwork, utility services
03000	Concrete	Includes scope itmes such as concrete reinforcement, concrete reinforcement
04000	Masonry	Includes scope itmes such as stone, basic masonry materials
05000	Metals	Includes scope itmes such as metal joists, metal deck, metal fabrications
06000	Wood plastics	Includes scope itmes such as rough carpentry, finish carpentry, basic wood and plaster
07000	Floorer	Includes scope itmes such as thermal protection, roofing and siding, flashing and sheet metal
08000	Doors and Windows	Includes scope itmes such as specialty doors, windows, skylights
09000	Finishes	Includes scope itmes such as tile, flooring, wall finishes
10000	Specialties	Includes scope itmes such as wall and corner guards, access flooring, storage shelving
11000	Equipment	Includes scope itmes such as instrumental equipment and office equipment
12000	Furnishings	Includes scope itmes such as fabrics, art, funiture
13000	Special Construction	Includes scope itmes such as swimming pools, storage tanks, suppression
14000	Conveying Systems	Includes scope itmes such as elevators, hoists and cranes,
15000	Mechanical	Includes scope itmes such as heating, ventilating, and air conditioning
16000	Electrical	Includes scope itmes such as lighting, communications, low voltage distribution

Table 8: Note that these categories come straight from the Construction Specifications Institute. I have provided a reference and link in Appendix A

CSI Codes

csi_code	name	description	qty_breakdown	csi_cat_id
03200	Concrete Reinforcement	Concrete Reinforcement	lsum	03000
03300	Cast In Place Concrete	Cast In place concrete	yard	03000
06100	Rough Carpentry	Rough carpentry includes raming and basic woodwork.	lsum	06000

Table 9: Note there are a lot more actual CSI Codes. This documentation shows a small sample of these codes.

Employees

employee_id	first_name	last_name	email	phone	emp_cat_id	project_count
1	Holly	Field	hollyf@homehelp.com	5032465578	3	NULL
2	Fred	Ferndale	fredf@homehelp.com	5032465576	1	1
3	Misty	May	mistym@homehelp.com	5032465574	1	1
4	Jim	Bean	jimb@homehelp.com	5032465572	4	NULL
5	Andrew	Landdog	andrewl@homehelp.com	5032465570	1	2
8	Ralph	Douglas	ralphd@homehelp.com	5032465568	5	4

Table 10: Employees example data. Note: Andrew Landdog is the account manager shown in the example above.

role_record_id	employee_id	project_id	role_cat_id	hours_worked
1	2	6	1	NULL
2	3	3	1	NULL
3	5	8	1	NULL
4	5	4	1	NULL
5	8	6	5	NULL
6	8	3	5	NULL
7	8	8	5	NULL
8	8	4	5	NULL

Table 11: Note this table works as an intersection table between the employees and projects. Note we have yet to utilized the hours worked function for employee work time.

Project Bids

bid_id	subcontractor_id	project_id	bid_amount	bid_approved
10001	1001	2	40000.00	0
10002	1002	2	41000.00	1
10003	1003	2	58000.00	1
10004	1002	4	42532.00	0
10005	1003	4	48080.00	0
10006	1005	4	36144.00	0
10007	1006	4	45903.00	0
10008	1008	4	34882.00	0
10009	1007	4	50148.00	0
10010	1003	4	49746.00	0
10011	1005	2	36000.00	0
10012	1003	8	NULL	0
10014	1002	8	NULL	0
10016	1003	8	NULL	0
10018	1003	8	NULL	0
10020	1003	8	NULL	0
10022	1007	8	NULL	0
10024	1007	8	32360.00	1
10026	1007	8	NULL	0
10027	1007	6	NULL	0
10028	1003	6	NULL	0
10029	1003	6	NULL	0
10030	1007	3	NULL	0
10031	1007	3	NULL	0
10033	1003	3	28000.00	0
10034	1002	6	28000.00	0
10035	1002	3	28000.00	0

10036	1002	8	28000.00	1
10037	1009	8	NULL	1

Table 12:Subcontractor Project Bids example data. Note-1: The bid approved column is a bit value showing 1 for TRUE and 0 for FALSE. Only the bids with a bid approved value of 1 are reported. The bids highlighted in light blue were used in the example report. Note-2: The yellow highlights illustrate how a subcontractor can bid on several projects using this table. Note-3: A redesign of this table could allow for one subcontractor to bid for multiple trades by including trade categories here.

scope_id	scope_name	csi_code	project_id	description
1	Concrete Foundation	03200	8	Project Requires new foundation
3	Concrete Foundation	03300	8	New foundation requires cast in place
4	Framing	06100	8	Project requires new framing

Table 13: This example table data details some of the scopes that are required for project with id of 8.

Subcontractors

subcontractor_id	name	street1	street2	city	state	zip	phone	email	project_count
1001	ABC Concrete	412 Peace St	NULL	Hillsboro	Or	97006	5032564111	service@abcconcrete.com	NULL
1002	Freds Concrete	8544 Busy Ave	NULL	Gresham	Or	97230	5038997526	fred@gmail.com	NULL
1003	Framers R Us	987 NW Ready St	NULL	Gresham	Or	97230	9712556333	info@framersrus.net	NULL
1004	A&B Electrical	555 Broadway Blvd	NULL	Portland	Or	97217	9715556444	jeremy@comcast.net	NULL
1005	Bob Ross	123 Trees Ave	NULL	Portland	Or	97215	5038994566	trees@bobspainting.com	NULL
1006	Jeremy's Pipe Services	54621 45th Ave	NULL	Portland	Or	97215	5034445621	service@jeremyspipes.com	NULL
1007	Portland Framers	121 Easy St	NULL	Portland	Or	97217	9712568999	info@portlandframers.com	NULL
1008	Lightning Electrical	184 NW 78th Blvd	NULL	Portland	Or	97217	9712626359	service@lightninge.com	NULL
1009	Rocks Concrete	404 Pine St	NULL	Portland	Or	97217	9715553456	dwayne@rockconcrete.com	NULL
1010	American Plumbing	5905 N Interstate Ave	NULL	Portland	Or	97217	5032896498	info@americanplumbing.com	NULL

Table 14: Subcontractors example data.

Appendix A – References

CSI Codes, www.mbionline.com, Construction Specifications Institute, www.mbionline.com/sites/default/files/public/documents/CSI_CODES.pdf.