

Design of the General Contractor Database for Suburban Remodeling Company

Prepared by Scott B. Bradley

For Partial Fulfillment of Requirements for

TS5501, Database Analysis and Design

Instructor: Dr. Wendy Waugh

March 15, 2004

Table of Contents

Introduction.....	3
Database Proposal Overview	4
Database Application Consideration.....	4
Deliverables	4
Introduction to the General Contractor Business System	5
Modeling the General Contractor Business System	5
Data Flow Diagrams (DFD).....	5
Use Case Diagrams (UCD).....	7
State Transition Diagrams.....	8
Sequence Diagrams.....	9
System Requirments	10
Developing the ER Diagram.....	13
The Data Dictionary.....	23
ADJUSTORS.....	25
BIDS	27
BID_PRICES	28
CUSTOMERS.....	30
ESTIMATES.....	32
EMPLOYEES	37
ESTIMATE_PRICES	38
INSURANCE_COS.....	40
INVOICES.....	43
ITEM_CATEGORIES.....	44
LABOR_TRADES	44
MATERIALS.....	44
PERMIT_CODES.....	46
PHOTOS.....	47
PRICELISTS.....	49
PRICELIST_PRICES	49
PROJECTS	51
PROJECT_NOTES.....	53
PURCHASES	54
SCHEDULES	56
STATES.....	58
STYLES.....	58
SUBS	59
VENDORS.....	61
ZIPs	64
SQL Code.....	65
Forms Design.....	86
SQL Query Examples	89
Conclusion	91
References.....	91

Introduction

As with many business systems, the amount of information tracked through business processes at Suburban Remodeling Company is quite large. This information includes, but is not necessarily limited to:

- Employees and employee payroll
- Business accounts
- Temporary employees
- Subcontractors and related contracts
- Customers (homeowners) and related project files
- Advertising information
- Insurance companies, estimate adjusters, and project related estimates and payouts
- Projects, including pending projects, projects sold, and projects closed
- Materials vendors and related invoices

With the primary exception of business accounts and payroll, which are tracked through the accounting program QuickBooks, most of the information within the Suburban Remodeling business system is tracked through paper trails and a variety of flat files such as Excel spreadsheets. Over the recent course of working with this data through the processes of analysis and data entry, I have gradually become aware of the complexity of the information tracked at Suburban Remodeling. I have also begun to learn of the relationship between that data and revenue earned at Suburban Remodeling Company. Inherent to the practice of general contracting is the business need for a keen awareness of the flow of materials and of monies related to those materials. In short, just as time is money in the business world, information is also valuable enough to be thought of as a form of currency. In the case of Suburban Remodeling Company, this is true particularly because so much revenue potential is gained or lost through the system of estimation and supplements.

Through these observations briefly summarized above, I have come to the following conclusion which is the proposal of this report: The business operations of Suburban Remodeling Company could benefit from a centralized data storage and retrieval system which I have the expertise to build. I have over ten years of experience in the application of engineering and information technology related solutions to a variety of industries. This experience has included building, configuring, and maintaining large databases capable of storing and manipulating very large quantities of information. I am also currently engaged in a program of graduate studies in Information Technology which has included the study of Information Technology Project Management, Systems Engineering, Systems Usability Analysis, and Database Analysis and Design.

Database Proposal Overview

My proposal is to create a database which would be used for the following:

- Facilitate the conversion of business data into electronic format through a customized database, modeled according to Suburban Remodeling Company's business operations, focusing primarily on the business processes related to project development and completion.
- Facilitate the tracking of information for project related business entities including customers, employees, subcontractors, insurance companies, and vendors
- Facilitate the analysis of this stored information through customized queries appropriate to business procedures and through the ability for ad hoc queries for future business needs.

Database Application Consideration

This database might be designed and built with any number of applications on the market. Herein, Microsoft Access will be considered to be the primary candidate for choice of applications because:

- Suburban Remodeling Company already owns Microsoft Access through licensed ownership of the Microsoft Office software suite
- Access runs in the familiar Microsoft Windows environment which is already used in the Suburban Remodeling office
- Access has inherent business-friendly capabilities such as the ability to design user-interface forms and the ability to create custom reports that can be exported to other Windows applications.
- Access has the capability of running in a server-client configuration like that used already in the Suburban Remodeling Company office.

Deliverables

Within the scope of the current proposal is the engineering of the proposed data system. This will be developed and presented in a manner toward anticipation of the eventual system delivery upon acceptance of the design to be proposed. Engineering plans will be presented in the form of a design specification which will include the following items related to the design of the database:

- Modeling synopsis of Suburban Remodeling Company's business processes, particularly with regards to project development and completion
- Data related business requirements associated with these processes
- Data related entity overview for this system
- A properly developed Entity Relationship Diagram, representing the logical data system

- A data dictionary derived from the E-R Diagram, which demonstrates data related information such as table names, attributes, data types, ranges, key value indicators, etc.
- A listing of SQL (Structured Query Language) necessary for the creation of the database tables appropriate to the proposed database application.
- A listing of SQL demonstrating example data queries appropriate to the business requirements and the proposed database.

Introduction to the General Contractor Business System

In the home building and remodeling industry, the general contractor serves to organize and to administer project related activities toward the end of a completed project. While the manual labor in the industry is procured by trade-specific subcontractors, the general contractor fulfills an important administrative role in the organization and completion of home building and remodeling projects.

This document begins the process of developing a comprehensive data system for Suburban Remodeling. (Note that the company name and sensitive data has been changed in this document in order to protect business interests.)

Modeling the General Contractor Business System

Toward the development of an operational data system to be integrated with the business practices of the general contractor, system models might be considered for the appropriation of our understanding of existing business practices into abstract form that might be translated into a working data system. The final data system, implemented as a relational database, will be based on an entity relationship diagram which will be the subject of later study. In this section, we will look at general approaches to business modeling and how they might be applied to the business of the general contractor.

The modeling examples to be demonstrated are part of the Unified Modeling Language (UML), the techniques of which might be applied to any business system. The UML is made up of several business modeling approaches which are translated into corresponding diagram types. The diagrams to be demonstrated here are: the Data Flow Diagram (DFD), the Use Case Diagram (UCD), the State Transition or Statechart diagram, and the Sequence Diagram.

Data Flow Diagrams (DFD)

One online source states the following about the Data Flow Diagram:

Data flow diagrams are useful in modeling high-level systems and processes. These diagrams show the way in which data flows between elements in a system and between the system and external entities. (Visualcase)

For the general contractor, the Data Flow Diagram will demonstrate the contracting business system as the core of a series of data flow by which the general contracting function operates. The emphasis in this diagram is on the flow of data between the primary entity (the general contractor) and related entities (sub-contractors, vendors, customers, etc.). Pfleeger emphasizes this data flow focus of the DFD,

The diagram shows the data that flow into the system, how they are transformed, and how they leave the system. The emphasis is always on the flow of data, not on the flow of control. (Pfleeger, 162).

In developing a systems model appropriate for the functions of a general contracting business, a high level model of data flow is important for recognizing the transformation of data between the contractor and external resources related to a project. Within the scope of the system to be defined, those external resources include the customer, the insurance company, the materials vendors, and the sub-contractors. Figure 1 demonstrates a high level Data Flow Diagram for the project related processes of the general contractor. This model relates directly to the data system to be developed for which the boundaries are defined by the project related functions of the general contractor, and for which inputs and outputs must be defined according to data flow between the system and related external resources.

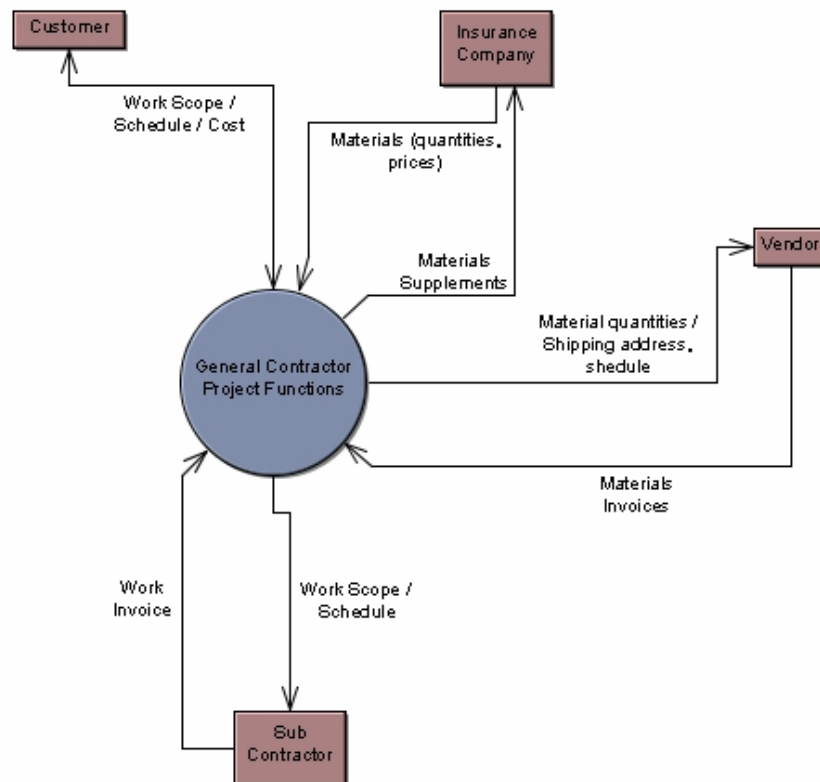


Figure 1. High level Data Flow Diagram.

Since these external resources will not be directly interacting with the database, the boundaries of the general system shown in this diagram include not only the database itself, but also the business system peripheral to the database, which includes the employees and the processes by which they enter data into and retrieve data from the database.

Use Case Diagrams (UCD)

The Use Case Diagram depicts the relationship between system users and system use cases. According to Sinan Si Alhir, in the book *UML in a Nutshell*,

Use case diagrams render the user view of a system. These diagrams describe the functionality provided by a system or class to external interactors. These diagrams contain actors, use cases, and their relationships. (Alhir, 159)

In the UCD, actors are not only human users, but may include other systems which utilize the system presented. Alhir states:

Actors are classes that define roles that objects external to a system may play. They are used to model users outside of a system that interact directly with the system as part of coherent work units. This includes human users and other systems. (Alhir, 160)

Actors may be indicated by a stereotyped class rectangle or as a stick person icon. Relationships generally depicted in the Use Case Diagram are the Communicates relationship, the Extends relationship and the Uses relationship. A summary definition of the UCD is provided by the Glossary of system modeling and development terms:

Use-case diagram: a diagram showing various external objects to the system (usually called actors in this context) and the functions they expect to see the system provide (sometimes called use cases in this context). The usual notation is to show each external object as a 'stick person' (though they can include hardware, system, and organisation object types, as well as 'people') and the functions as ellipses within the system, which is represented as a rectangle. (Glossary)

Applied to the general contractor system, the diagram depicts the users and the use cases shown below in Table 1. The UCD of this system is shown in figure 2.

User	Use Cases
Sales Representative	Customer processes
Estimator	Customer processes, Insurance processes
Project Manager	Customer processes, Insurance processes, Vendor processes, Subcontractor processes
Office Administrator	Customer processes, Insurance processes, Vendor processes, Subcontractor processes

Table 1. UCD Relationships for the General Contractor System

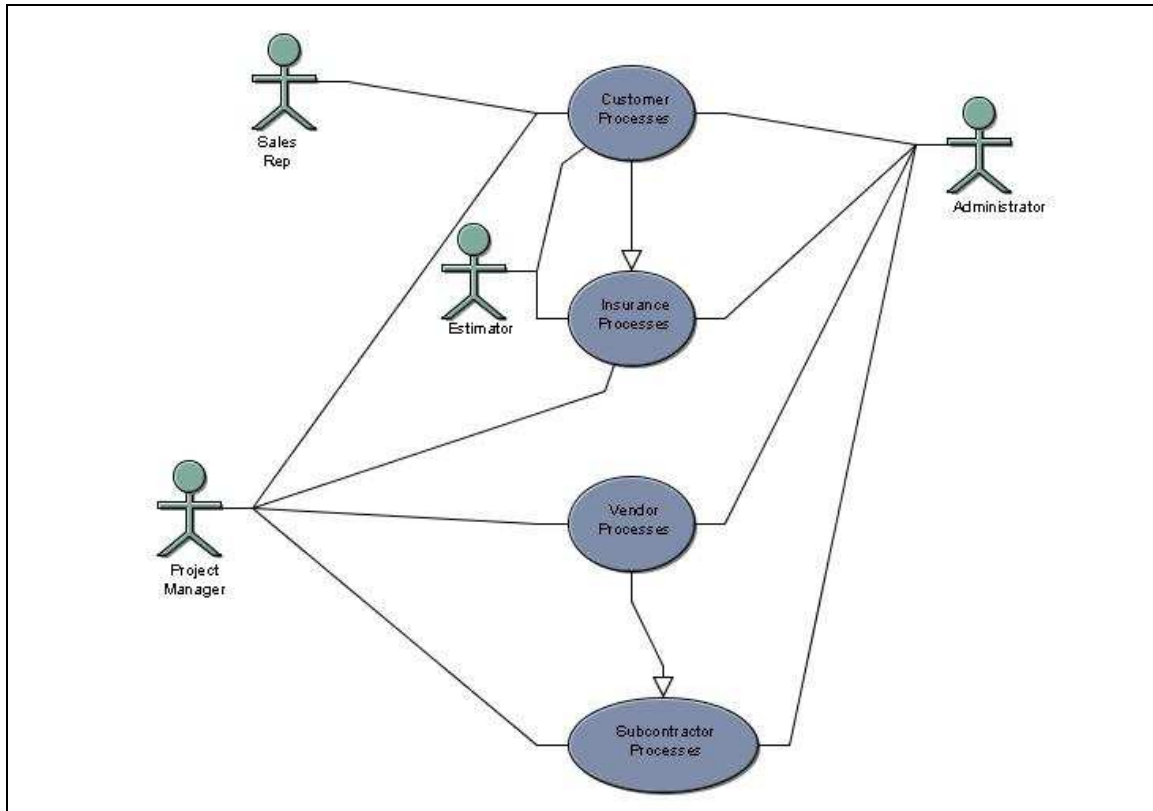


Figure 2. Use Case Diagram for the General Contractor Business System

In the UCD as depicted, the actors in this system represent the users who will be utilizing the data inherent to the use case processes. In developing a database model, these use cases will form the beginning of developing entities for the sake of an overall business system model. For example, the high level use case shown here of “Vendor Processes” might translate into such related data entities as Vendor and Product, while “Subcontractor Processes” implies such related entities as Subcontractor and Trade.

State Transition Diagrams

From the glossary: “**State transition diagram**: a modelling tool that highlights the way something changes over time (its dynamics).” (Glossary)

And, Alhir gives us further insight:

[State] diagrams lie within the behavioral view of a system and render the states and responses of a class. These diagrams describe the behavior of a class in response to external stimuli. [State]

diagrams describe the life cycle of an object. These diagrams contain states and transitions. (Alhir, 187)

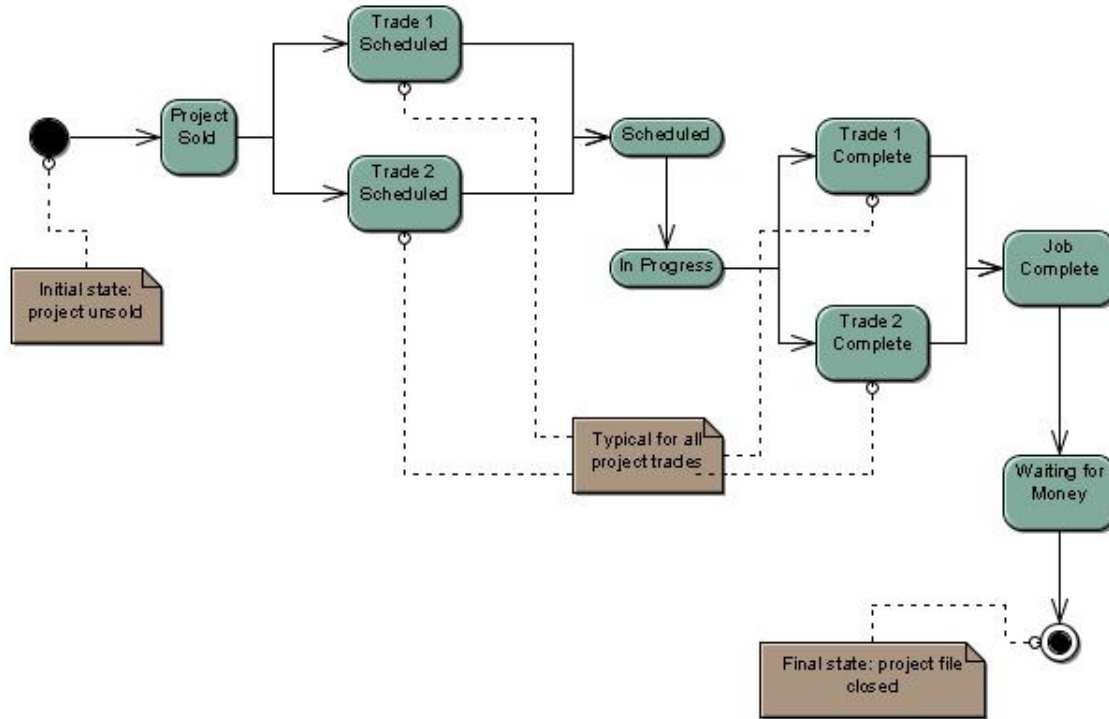


Figure 3. Statechart Diagram of Project Status

For the general contractor business system, the most significant business-oriented life cycle is that of the project. The term project here entails the start to finish process of acquiring, scheduling, completing, and closing a home remodeling project. This life cycle is depicted in the statechart diagram of Figure 3. This diagram gives us an additional understanding of the overall system, further enabling the development of a data system model. From the statechart diagram shown, entities such as Project, Trade, and Schedule might be derived. Furthermore, the diagram is indicative of such related entity attributes as Project Status, and Schedule Date.

Sequence Diagrams

According to Alhir,

Sequence diagrams lie within the behavioral view of a system and render the specification of behavior. These diagrams describe the behavior provided by a system to interactors. (Alhir, 167)

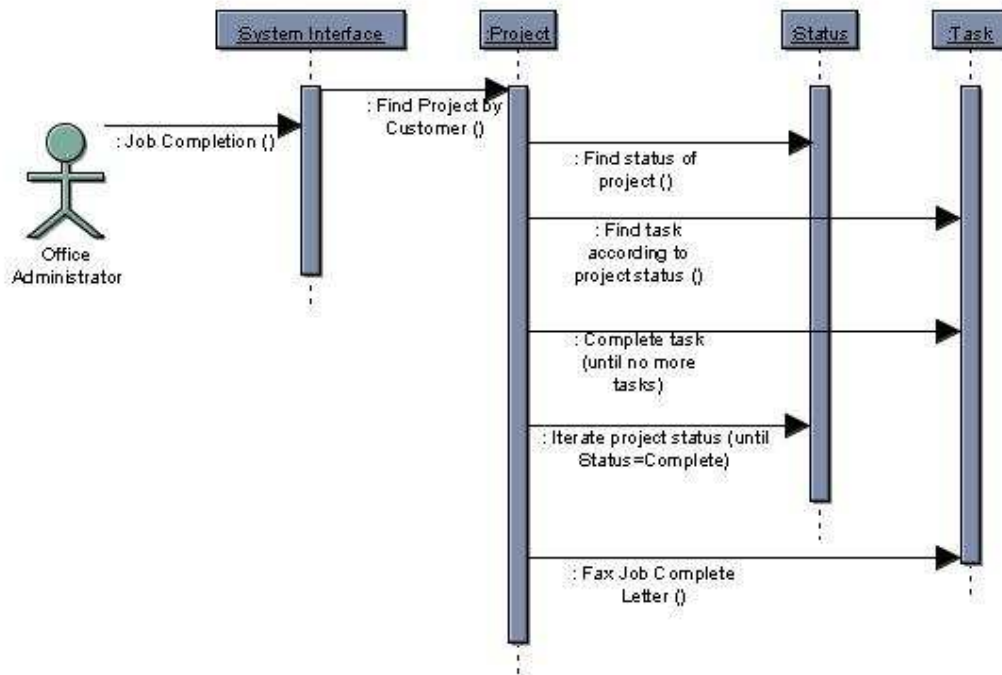


Figure 4. Sequence Diagram of Job Completion

Sequence diagrams detail very specific operations within the overall business system. Although detailing all sequence processes within the overall business system can be a labor intensive process, such detail may be necessary for certain key procedures. Figure 4 demonstrates an example of the applicability of the Sequence Diagram to the general contractor business system. This particular sequence depicts the business process flow toward achieving project completion, and demonstrates the particular system output of a “Job complete fax,” as a part of that sequence process.

System Requirements

From this brief discussion of modeling the general contractor business system, we can determine a number of implications for the requirements of our system. It is easily seen that the remodeling business processes are intrinsically linked with the life cycle of the home remodeling project. For this reason, the project is the center of focus for the overall system. Within a given project, then, there are many discernible business entities that work together toward the completion of the remodeling project life cycle. Those entities include both internal and external entities, and with each is associated a set of data appropriate to the entity’s sub-process. These entities and their respective data will be further developed in the next section on the development of the Entity Relationship diagram.

An important consideration in the mapping of the overall data system is that of existing data resources. Suburban Remodeling uses a particular estimating program called Xactimate which plays a significant role in shaping the overall project life cycle. Fortunately, much of the important information available in Xactimate is also able to be exported for use in other applications. The current database design will exploit this capability in order to better integrate itself with existing business processes.

The system will therefore work closely with the Xactimate program which is used to track estimating prices. Xactimate provides a number of data dumps which may be used for direct integration of Xactimate and the overall information system. The tables below demonstrate samples of this data available from Xactimate. Table 1 shows a sample of material/labor categories each with a 3 character Code ID and up to a 30 character description. Table 2 shows a sample of material items data available from Xactimate. This data includes a unique item ID of up to 10 characters, including the 3 character category code prefix. Also included are a description, a unit, and a unit price.

Code	Description
PLA	INTERIOR LATH & PLASTER
PLM	PLUMBING
PNL	PANELING & WOOD WALL FINISHES
PNT	PAINTING
POL	SWIMMING POOLS & SPAS
RFG	ROOFING
SCF	SCAFFOLDING
SDG	SIDING
SFG	SOFFIT, FASCIA, & GUTTER
SPE	SPECIALTY ITEMS
STJ	STEEL JOIST COMPONENTS
STL	STEEL COMPONENTS
STR	STAIRS

Table 2. Xactimate Category List Sample Data

Item Code	Description	Unit	Unit Price
RFGWSHKR	Wood shakes - medium hand split - fire resistant	SQ	391.98
RFGWSHKR+	Wood shakes - heavy hand split - fire resistant	SQ	394.71
RFGWSHKRE	Wood shakes - medium hand split - fire resist. (per SHAKE)	EA	10.4
RFGWSHKRE+	Wood shakes - heavy hand split - fire resist. (per SHAKE)	EA	10.41
RFGWSRMV	Tear off, haul and dispose of wood shingles	SQ	43.55
RFGWSRMVN	Tear off wood shingles (no haul off)	SQ	35.73
RFGWSTP	Wood shingles - taper sawn - #1 cedar	SQ	360.82
RFGWSTPE	Wood shingles - taper sawn - #1 Cedar (per SHINGLE)	EA	10.32
RFGWSTPR	Wood shingles - taper sawn - fire resistant - #1 cedar	SQ	384.39

Table 3. Xactimate Item List Sample Data

With the above models and business facts in mind, we can begin to assess the needed criteria for a business information database system. The most important criteria to be considered follow:

- The system should allow for the tracking of individual projects and their associated customers, sales representatives, insurance companies, estimates, schedules, etc.
- Provisions should be made for tracking material prices by (1) internal estimates (bids), (2) external estimates (scopes), and (3) industry standard pricelists
- The system should facilitate the organization and retrieval of project related digital photographs
- Contact management should be maintained for all individuals within the business system, including employees, subcontractors, vendors, insurance adjustors and customers.
- Detailed invoicing and price tracking should be facilitated, including the comparison of scope versus purchase prices for given materials, and the comparison of scope payouts between insurance companies over time.
- The system should provide for the tracking of subcontractors, their related fields of specialization, and there work schedules in relation to active projects.
- Provisions should be made for tracking project related details including project status, and various notes related to project activities, as well as tracking employees responsible for the project notes.
- The system should contain information about building permits which maybe required according to geographic area and building trade.
- Post-project completion payouts should be tracked, including the status of payments for overhead and profit, PWI, and depreciation withheld.

Developing the ER Diagram

Arguably, the most important model to develop in the process of database modeling and development is the Entity Relationship Diagram. As its name implies, this diagram models the system in terms of its entities and their many relationships. As we develop the Entity Relationship model for the business processes of the general contractor, the models discussed in this section will serve as significant touchstones for the development of an overall understanding of system entity relationships.

Entities which have been selected to model the business processes of Suburban Remodeling Company are as follows:

Adjustor	Photo
Bid	Pricelist
Bid Prices	Pricelist Price
Customer	Project
Employee	Project Note
Estimate	Purchase
Estimate Price	Schedule
Insurance Company	State
Invoice	Style
Labor Trades	Sub
Material	Vendor
Permit Code	ZIP

The resulting Entity Relationship Diagram is shown:

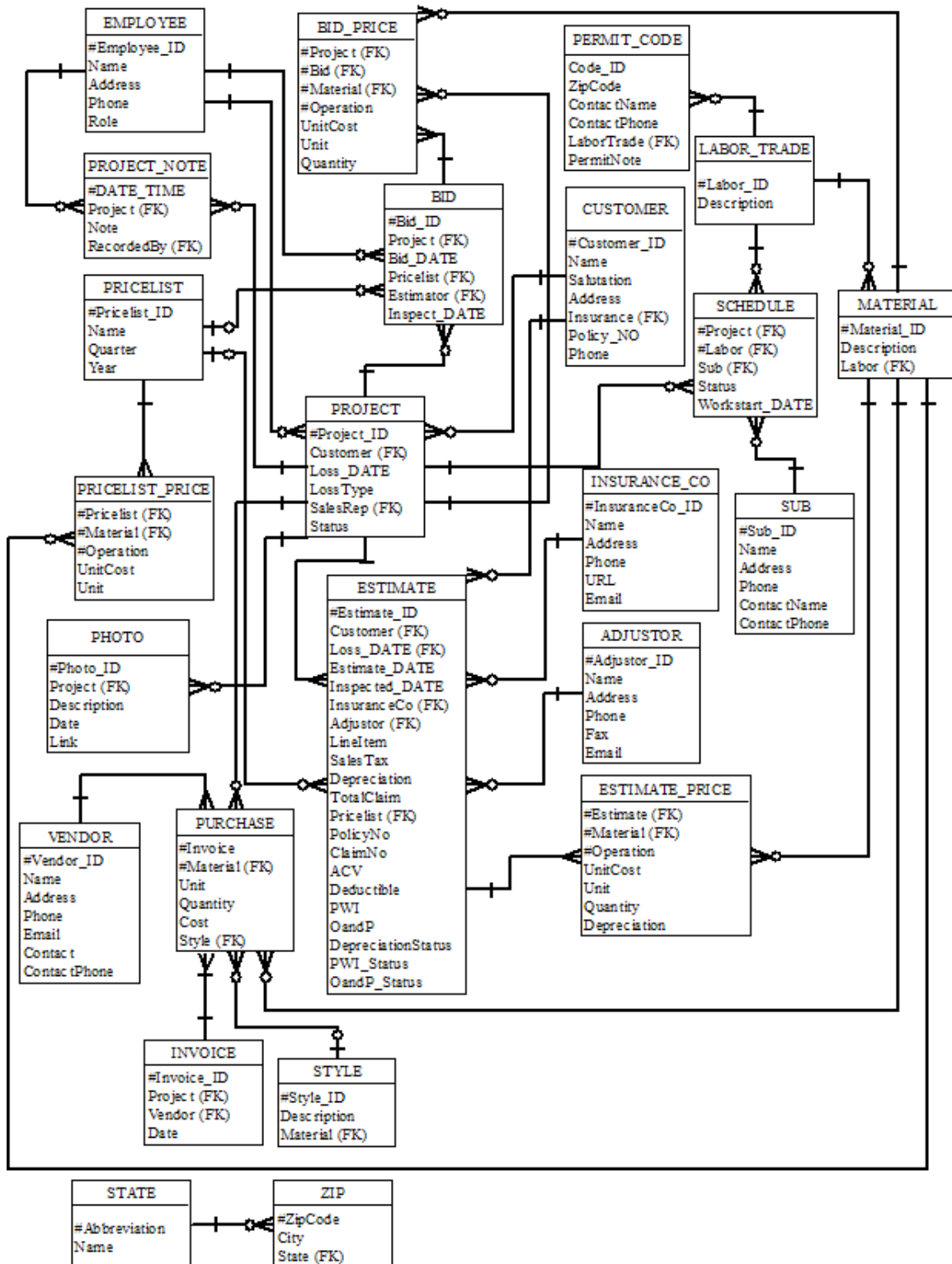


Figure 4. Entity Relationship Diagram for the General Contractor Database

It is important to note that the demonstrated ER diagram represents merely the beginning of the design process. Since part of the intent of this document is to demonstrate the design process in action, the ER diagram shown does not necessarily reflect the final database. Several Attributes have been removed for the sake of simplicity (allowing the diagram to fit on a single page). Commonly referenced entities (STATES and ZIPS) have been shown without relationships as their relationships with other entities are frequent and are generally obvious. These simplified elements of the overall design will be worked back into the design as the data dictionary and corresponding SQL are developed. Furthermore, as the design process develops through the following pages, further reflection and design refinement will require the fine tuning of design elements. Where thought to offer insight into the design process, the motivations of such changes will be noted. The final implementation of the database should be understood to be the data dictionary and the SQL table creation code later in this document as they represent the database implementation which has gone through the engineering process as well as the building of a testing/prototype database.

To demonstrate normalization of the current database schema, dependency diagrams follow. For illustrative purposes, the normalization process is shown where appropriate through the use of preliminary and corrected dependency diagrams and appropriate notes. The ER diagram above reflects only the final normalized form.

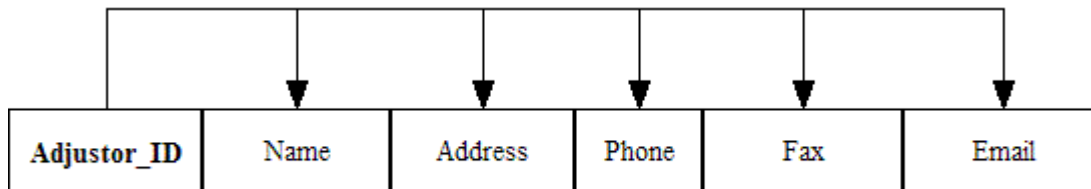
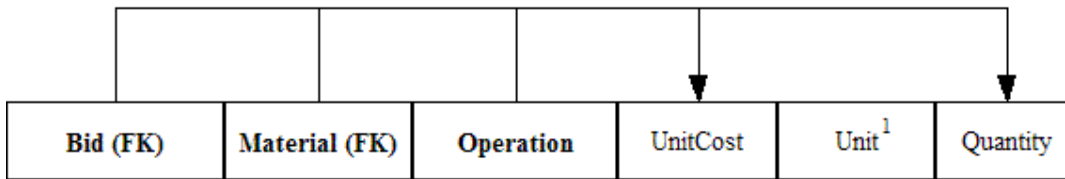


Figure 5. ADJUSTORS entities dependencies.

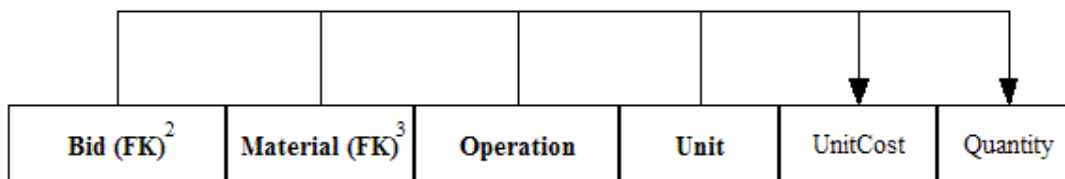
Original BID_PRICES table



1. The same material may be called out at various units, e.g. roofing may be estimated per shingle or per square (100 SF). Therefore, in this representation, the BID_PRICES table is in violation of 1NF. Possible solutions considered follow:
 - Move the Unit attribute to the MATERIALS table, creating distinct material rows in cases where a material requires reference by various units.
 - Make Unit part of the compound primary key.

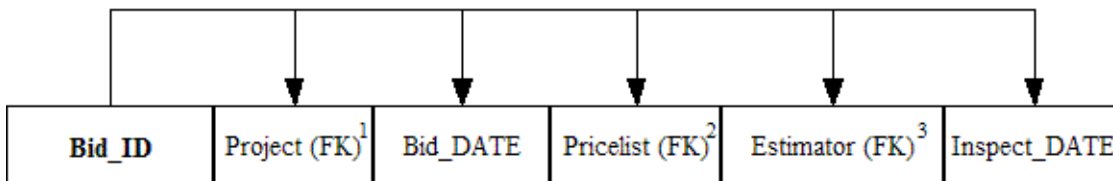
Since the first solution requires creating unnecessary additional material rows, and would further complicate the reporting of materials for cost, estimate, and usage comparisons, the latter solution will be used, resulting in the corrected dependency diagram shown below.

Corrected BID_PRICES table



2. Bid references BIDS (Bid_ID)
3. Material references MATERIALS (Material_ID)

Figure 6. BID_PRICES entities dependencies.



1. Project references PROJECTS (Project_ID)
2. Pricelist references PRICELISTS (Pricelist_ID)
3. Estimator references EMPLOYEE (Employee_ID)

Figure 7. BIDS entities dependencies.

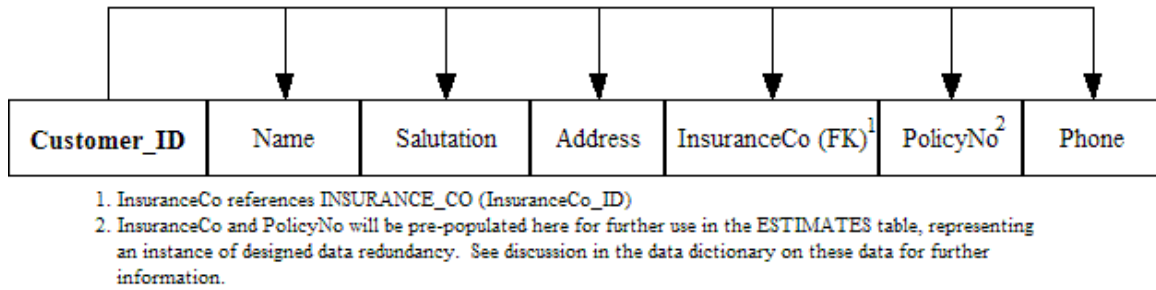


Figure 8. CUSTOMERS entities dependencies.

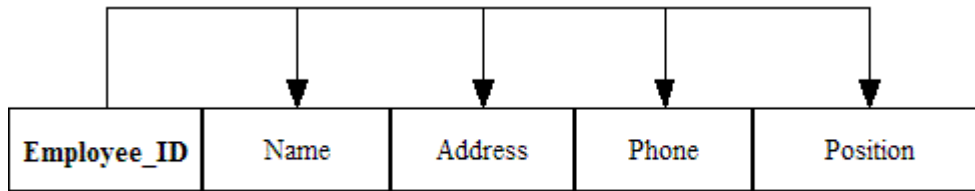


Figure 9. EMPLOYEES entities dependencies.

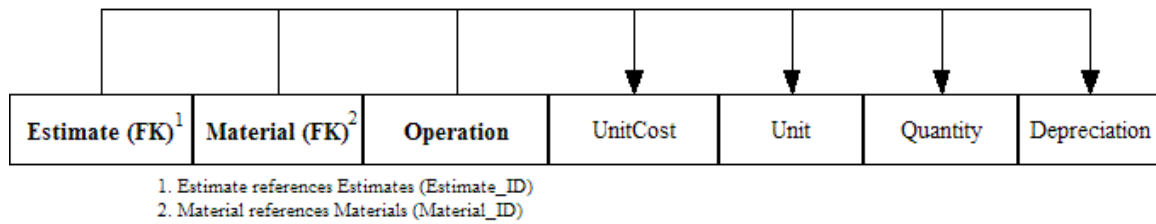


Figure 10. ESTIMATE_PRICES entities dependencies.

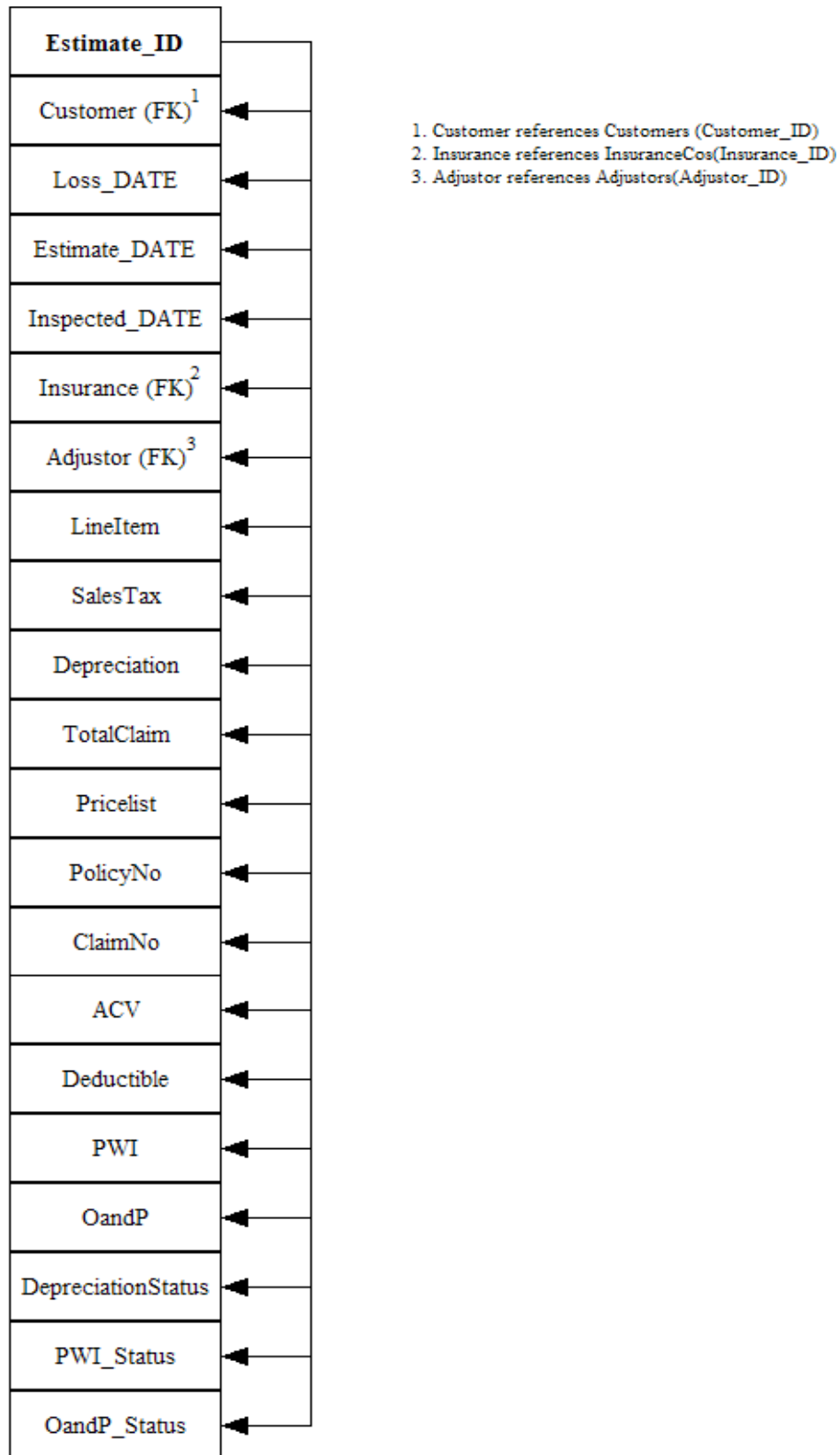


Figure 11. ESTIMATES entities dependencies.

Copyright © 2004 by Scott B. Bradley

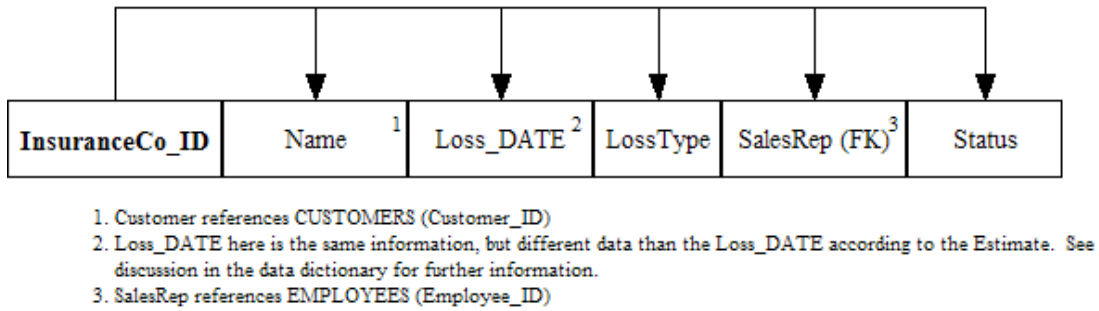


Figure 12. INSURANCE_COS entities dependencies.

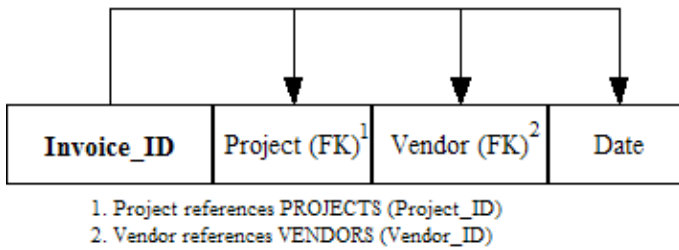


Figure 13. INVOICES entities dependencies.

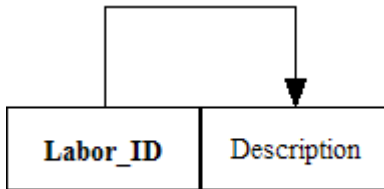


Figure 14. LABOR_TRADES entities dependencies.

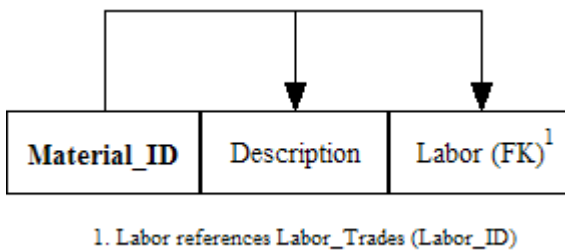


Figure 15. MATERIALS entities dependencies.

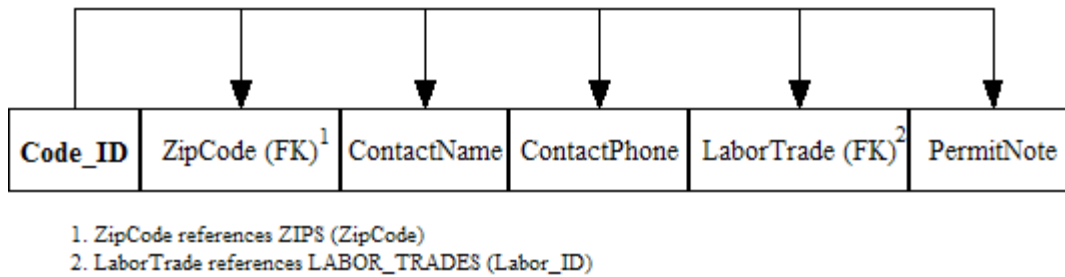


Figure 16. ZIPS entities dependencies.

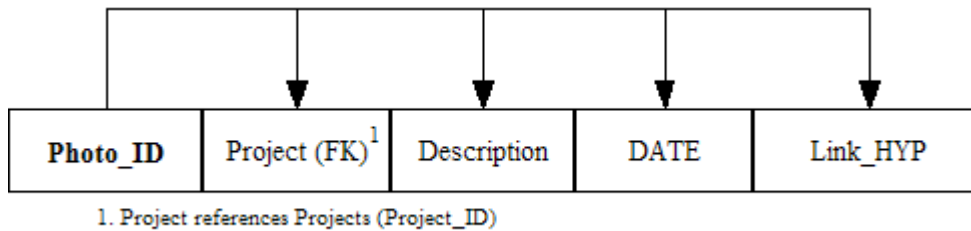


Figure 17. PHOTOS entities dependencies.

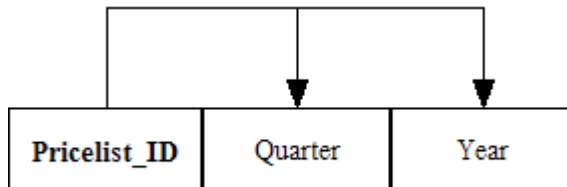


Figure 18. PRICELISTS entities dependencies.

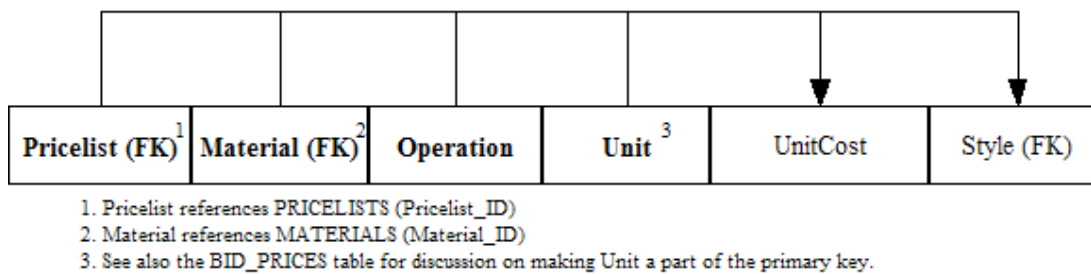
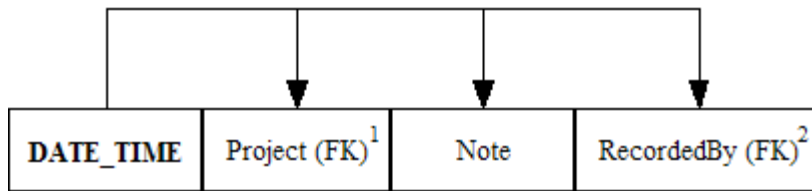
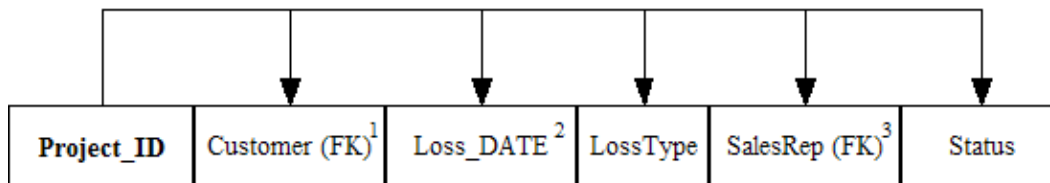


Figure 19. PRICELISTS entities dependencies.



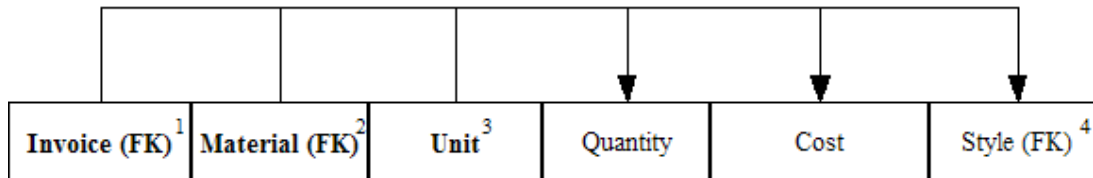
1. Project references Projects (Project_ID)
2. RecordedBy references Employees (Employee_ID)

Figure 20. PROJECT_NOTES entities dependencies.



1. Customer references CUSTOMERS (Customer_ID)
2. Loss_DATE here is the same information, but different data than the Loss_DATE according to the Estimate. See discussion in the data dictionary for further information.
3. SalesRep references EMPLOYEES (Employee_ID)

Figure 21. PROJECTS entities dependencies.



1. Invoice references INVOICES (InvoiceNo)
2. Material references MATERIALS (Material_ID)
3. See also the BID_PRICES table for discussion on making Unit a part of the primary key.
4. Style references STYLES (Style_ID)

Figure 22. INVOICES entities dependencies.

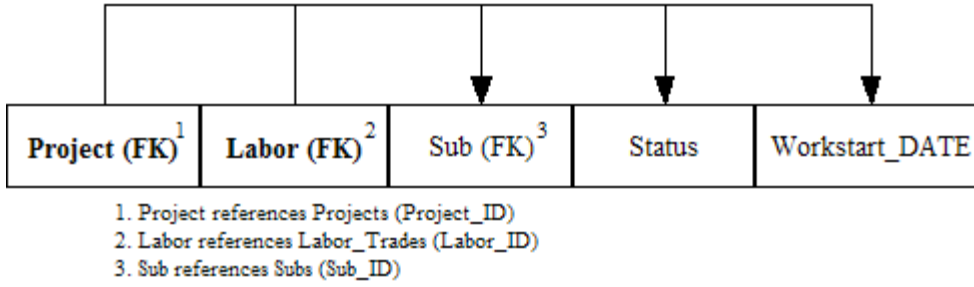


Figure 23. SCHEDULES entities dependencies.

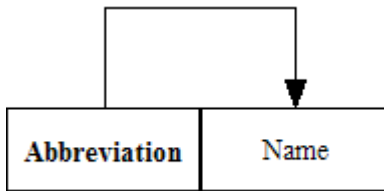


Figure 24. STATES entities dependencies.

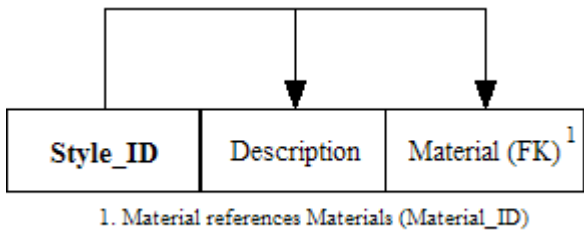


Figure 25. STYLES entities dependencies.

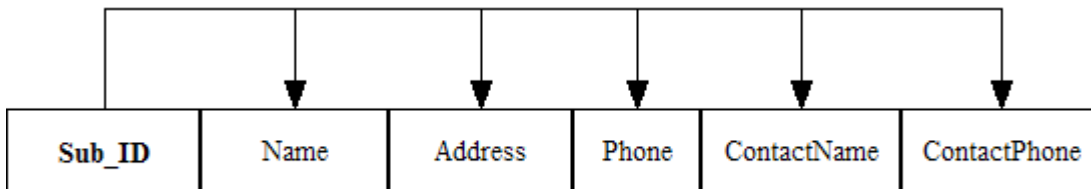


Figure 26. SUBS entities dependencies.

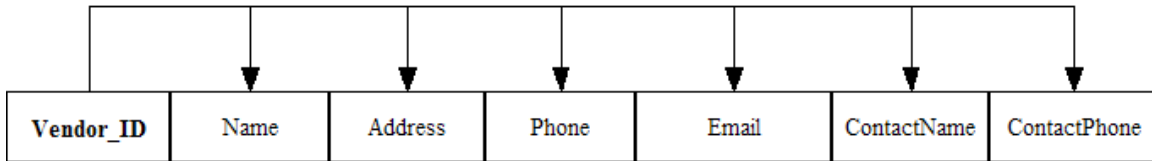


Figure 27. VENDORS entities dependencies.

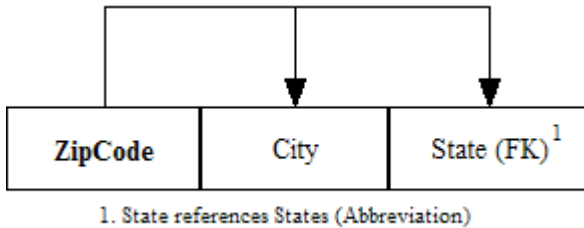


Figure 28. PROJECT_NOTES entities dependencies.

The Data Dictionary

Based on the given models for business processes at Suburban Remodeling Company, a data dictionary may be developed. This dictionary will provide the basis for constructing the business information database, and will also serve as a useful reference tool during database administration and maintenance.

Dictionary key

Table Name

Name of table. By convention, table names will be indicated in all CAPITALS and will be pluralized according to the multiple rows they will contain.

Attributes

A summary of the attributes to follow. # indicates primary key values.

Attribute Name

Attribute corresponding to a column of data. Naming conventions include:

Where non-text attributes occur, for clarity the attribute type may be indicated with a suffix in CAPS such as _DATE or _NUMBER. This convention is followed only where it serves to clarify and is a natural part of the attribute name.

Attributes will be named in Mixed Case for clarity. Underscores will be used only where mixed case cannot provide sufficient clarity.

Contents

Description of field data.

Type

Data type. For Microsoft Access, the following data types are acceptable:

text

memo

yes/no

number (byte)

number (integer)

number (long integer)

number (single)

number (double)

number (Replication ID)

number (Decimal)

date/time

Currency

AutoNumber (Long Integer)

AutoNumber (Replication ID)

OLE Object

Hyperlink

Format

Formatting characteristics of the field data.

Range

Range of values the field may contain.

Required

Yes if null values are not allowed.

Key

PK indicates primary key; FK and reference indicates a foreign key and the referenced field. Foreign key references are in the form TABLE (field).

ADJUSTORS

Insurance adjustor responsible for creating and providing an insurance scope/estimate.

ADJUSTORS (Adjustor_ID)

Attribute	Contents			
#Adjustor_ID	Numeric key value			
Type	Format	Range	Required	Key
AutoNumber (long integer)			Yes	PK

ADJUSTORS (FirstName)

Attribute	Contents			
FirstName	First name of the insurance adjustor			
Type	Format	Range	Required	Key
Text			No	

ADJUSTORS (LastName)

Attribute	Contents			
LastName	Last of the insurance adjustor			
Type	Format	Range	Required	Key
Text			No	

ADJUSTORS (Address)

Attribute	Contents			
Address	Adjustor's business address number and street			
Type	Format	Range	Required	Key
Text			No	

ADJUSTORS (ZipCode)

Attribute	Contents			
ZipCode (FK)	Zip code associated with the address			
Type	Format	Range	Required	Key
Text	#####		No	FK ZIPS(ZipCode)

ADJUSTORS (PhoneAreaCode)

Attribute	Contents			
PhoneAreaCode	Area code associated with adjustor's phone number			
Type	Format	Range	Required	Key
Text	###	000-999	No	

ADJUSTORS (PhoneNumber)

Attribute	Contents			
PhoneNumber	Adjustor's 7 digit phone number			
Type	Format	Range	Required	Key
Text	###-####	000-0000 to 999-9999	No	

ADJUSTORS (FaxAreaCode)

Attribute	Contents			
FaxAreaCode	Area code associated with adjustor's fax number			
Type	Format	Range	Required	Key
text	###	000-999	No	

ADJUSTORS (FaxNumber)

Attribute	Contents			
FaxNumber	Adjustor's 7 digit fax number			
Type	Format	Range	Required	Key

Text	###-####	000-0000 to 999-9999	No	
------	----------	-------------------------	----	--

ADJUSTORS (Email)

Attribute	Contents			
Email	Adjustor's email address			
Type	Format	Range	Required	Key
HYP				

BIDS

Internal estimates created by estimating specialist.

BIDS (Bid_ID)

Attribute	Contents			
#Bid_ID	Numeric key value			
Type	Format	Range	Required	Key
AutoNumber (long integer)			Yes	PK

BIDS (Project)

Attribute	Contents			
Project	Project identifier			
Type	Format	Range	Required	Key
Number (long integer)			Yes	FK PROJECTS (Project_ID)

BIDS (Bid_DATE)

Attribute	Contents
Bid_DATE	Date the bid was created

Type	Format	Range	Required	Key
Date/Time			Yes	

BIDS (Pricelist)

Attribute	Contents			
Pricelist	Identifier of Pricelist the bid prices were based on.			
Type	Format	Range	Required	Key
Text			No	FK PRICELISTS (Pricelist_ID)

BIDS (Estimator)

Attribute	Contents			
Estimator	Identifier of employee responsible for creating the bid.			
Type	Format	Range	Required	Key
Number (long integer)			No	FK EMPLOYEES (Employee_ID)

BIDS (Inspected_DATE)

Attribute	Contents			
Inspected_DATE	Date the home was measured for estimating.			
Type	Format	Range	Required	Key
Date/Time			No	

BID_PRICES

Line item materials with associated quantities and prices for an internal estimate/bid.

BID_PRICES (Bid)

Attribute	Contents			
#Bid (FK)	Identifier of associated bid.			
Type	Format	Range	Required	Key
Number (Long Integer)			Yes	PK, FK BIDS (Bid_ID)

BID_PRICES (Category)

Attribute	Contents			
#Category (FK)	3 character item category code corresponding to Xactimate codes.			
Type	Format	Range	Required	Key
Text	XXX		Yes	PK, FK MATERIALS(Category)

BID_PRICES (Material)

Attribute	Contents			
#Material (FK)	Identifier of material priced			
Type	Format	Range	Required	Key
Text			Yes	PK, FK Materials (Material_ID)

BID_PRICES (UnitPrice)

Attribute	Contents			
UnitPrice	Line item bid price.			
Type	Format	Range	Required	Key
Currency	#.##		Yes	

BID_PRICES (Quantity)

Attribute	Contents			
Quantity	Quantity of material estimated to be required.			

Type	Format	Range	Required	Key
Number (Long)	#.##		Yes	

CUSTOMERS

Homeowners of homes being remodeled.

CUSTOMERS (Customer_ID)

Attribute	Contents			
#Customer_ID	Numeric key value.			
Type	Format	Range	Required	Key
AutoNumber (Long Integer)			Yes	PK

CUSTOMERS (FirstName)

Attribute	Contents			
FirstName	First name of primary customer contact.			
Type	Format	Range	Required	Key
Text			Yes	

CUSTOMERS (LastName)

Attribute	Contents			
LastName	Last name of primary customer contact.			
Type	Format	Range	Required	Key
Text			Yes	

CUSTOMERS (Salutation)

Attribute	Contents
-----------	----------

Salutation	Preferred greeting for mailing purposes.			
Type	Format	Range	Required	Key
Text			No	

CUSTOMERS (Address)

Attribute	Contents			
Address	Customer's primary contact street name and number.			
Type	Format	Range	Required	Key
Text			Yes	

CUSTOMERS (ZipCode)

Attribute	Contents			
ZipCode (FK)	Zip Code associated with customer address.			
Type	Format	Range	Required	Key
Text	#####	00000-99999	Yes	FK ZIPS(ZipCode)

CUSTOMERS (InsuranceCo)

Attribute	Contents			
InsuranceCo (FK)	Key identifier of the customer's insurance company.			
Type	Format	Range	Required	Key
Text			No	FK INSURANCE_COS (InsuranceCo_ID)

CUSTOMERS (PolicyNo)

Attribute	Contents			
PolicyNo	Customer's Insurance policy number.			
Type	Format	Range	Required	Key
Text			No	

CUSTOMERS (PhoneAreaCode)

Attribute	Contents			
PhoneAreaCode	Area code associated with customer's phone number.			
Type	Format	Range	Required	Key
Text	###	000 to 999	No	

CUSTOMERS (PhoneNumber)

Attribute	Contents			
PhoneNumber	Customer's 7 digit phone number.			
Type	Format	Range	Required	Key
Text	###-####	000-0000 to 999-9999	No	

ESTIMATES

Remodeling cost estimate provided by the insurance company or third party estimator. Also called a scope.

ESTIMATES (Estimate_ID)

Attribute	Contents			
#Estimate_ID	Numeric key value			
Type	Format	Range	Required	Key
AutoNumber (Long Integer)			Yes	PK

ESTIMATES (Customer)

Attribute	Contents			
Customer (FK)	Numeric identifier of the customer associated with the estimate.			
Type	Format	Range	Required	Key
Number (Long integer)			Yes	FK CUSTOMERS (Customer_ID)

ESTIMATES (Loss_DATE)

Attribute	Contents			
Loss_DATE	Date damage to home occurred.			
Type	Format	Range	Required	Key
Date/Time			No	

ESTIMATES (Estimate_DATE)

Attribute	Contents			
Estimate_DATE	Date estimate was created.			
Type	Format	Range	Required	Key
Date/Time			No	

ESTIMATES (Inspected_DATE)

Attribute	Contents			
Inspected_DATE	Date home was inspected by insurance adjustor.			
Type	Format	Range	Required	Key
Date/Time			No	

ESTIMATES (Insurance)

Attribute	Contents			
Insurance (FK)	Key identifier of insurance company associated with this scope.			
Type	Format	Range	Required	Key
Text			No	FK INSURANCE_COS (InsuranceCo_ID)

ESTIMATES (Adjustor)

Attribute	Contents			
Adjustor (FK)	Numeric identifier of the adjustor.			

Type	Format	Range	Required	Key
Number (Long Integer)			No	FK ADJUSTORS (Adjustor_ID)

ESTIMATES (SalesTax)

Attribute	Contents			
SalesTax	Sales tax percentage applied to scope in multiplier format.			
Type	Format	Range	Required	Key
Number	0.###	0.00 to 0.99	No	

ESTIMATES (Depreciation)

Attribute	Contents			
Depreciation	Total depreciation of home withheld from initial insurance payment.			
Type	Format	Range	Required	Key
Currency	#.###		No	

ESTIMATES (TotalClaim)

Attribute	Contents			
TotalClaim	Total value of insurance claim.			
Type	Format	Range	Required	Key
Currency	#.###		No	

ESTIMATES (Pricelist)

Attribute	Contents			
Pricelist	Name of pricelist used to create scope.			
Type	Format	Range	Required	Key
Text			No	*

* Unlike an internal bid for which the originating pricelist can be known for certain and thus referenced as a foreign key with integrity, the value here is based only on a name identifier supplied on the insurance scope. Thus a foreign key is not utilized.

ESTIMATES (PolicyNo)

Attribute	Contents			
PolicyNo				
Type	Format	Range	Required	Key
			No	

ESTIMATES (ClaimNo)

Attribute	Contents			
ClaimNo	Identifier of insurance claim associated with the scope.			
Type	Format	Range	Required	Key
Text			No	

ESTIMATES (ACV)

Attribute	Contents			
ACV	Total actual cash value.			
Type	Format	Range	Required	Key
Currency	###		No	

ESTIMATES (PWI)

Attribute	Contents			
PWI	PWI withheld from initial insurance payment.			
Type	Format	Range	Required	Key
Currency	###		No	

ESTIMATES (Deductible)

Attribute	Contents			
Deductible	Amount of insurance deductible for which the homeowner is responsible			
Type	Format	Range	Required	Key

Currency	###		No	
----------	-----	--	----	--

ESTIMATES (OandP)

Attribute	Contents			
OandP	Total overhead and profit to be paid on project completion.			
Type	Format	Range	Required	Key
Currency	###		No	

ESTIMATES (DepreciationStatus)

Attribute	Contents			
DepreciationStatus	Status of payment of depreciation withheld.			
Type	Format	Range	Required	Key
Text			No	

ESTIMATES (PWI_Status)

Attribute	Contents			
PWI_Status	Status of payment of PWI withheld.			
Type	Format	Range	Required	Key
Text			No	

ESTIMATES (OandP_Status)

Attribute	Contents			
OandP_Status	Status of payment of overhead and profit withheld.			
Type	Format	Range	Required	Key
Text			No	

EMPLOYEES

Internal employees, their contact information and primary role in the company.

EMPLOYEES (Employee_ID)

Attribute	Contents			
#Employee_ID	Numeric key value.			
Type	Format	Range	Required	Key
AutoNumber (Long Integer)			Yes	PK

EMPLOYEES (FirstName)

Attribute	Contents			
FirstName	First name of employee			
Type	Format	Range	Required	Key
Text			Yes	

EMPLOYEES (LastName)

Attribute	Contents			
LastName	Last name of employee			
Type	Format	Range	Required	Key
Text			Yes	

EMPLOYEES (Address)

Attribute	Contents			
Address	Employees primary street name and number.			
Type	Format	Range	Required	Key
Text			No	

EMPLOYEES (ZipCode)

Attribute	Contents			
ZipCode (FK)	Zip Code associated with employee address			
Type	Format	Range	Required	Key
Text	#####	00000-9999	No	FK ZIPS(ZipCode)

EMPLOYEES (PhoneNumber)

Attribute	Contents			
Phone	Employees primary 7 digit phone number			
Type	Format	Range	Required	Key
Text	###-####	000-0000 to 999-9999	No	

EMPLOYEES (PhoneAreaCode)

Attribute	Contents			
PhoneAreaCode	Area code associated with employee's primary phone number.			
Type	Format	Range	Required	Key
Text	###	000 to 999	No	

EMPLOYEES (Role)*

Attribute	Contents			
Role	Description of employee's primary role in the company.			
Type	Format	Range	Required	Key
Text			No	

*Because POSITION is an SQL reserved word, this attribute name has been changed to Role.

ESTIMATE_PRICES

Line item prices for insurance estimates including materials, quantities and values.

ESTIMATE_PRICES (Estimate)

Attribute	Contents			
#Estimate (FK)	Numeric identifier of associated insurance estimate/scope.			
Type	Format	Range	Required	Key
Number (Long Integer)			Yes	PK, FK ESTIMATES (Estimate_ID)

ESTIMATE_PRICES (Category)

Attribute	Contents			
#Category	3 character trade category identifier.			
Type	Format	Range	Required	Key
Text	XXX		Yes	PK, FK MATERIALS (Category)

ESTIMATE_PRICES (Material)

Attribute	Contents			
#Material	Text identifier of line item material.			
Type	Format	Range	Required	Key
Text			Yes	PK, FK MATERIALS (Material_ID)

ESTIMATE_PRICES (Operation)

Attribute	Contents			
#Operation	Labor operation associated with item estimate.			
Type	Format	Range	Required	Key
Text		Value in: Remove Replace R&R Det&Rst	Yes	PK

ESTIMATE_PRICES (UnitCost)

Attribute	Contents			
UnitCost	Line item estimate price.			
Type	Format	Range	Required	Key
Currency	#.##		Yes	

ESTIMATE_PRICES (Quantity)

Attribute	Contents			
Quantity	Quantity of material estimated to be required.			
Type	Format	Range	Required	Key
Number	#.##		Yes	

ESTIMATE_PRICES (Depreciation)

Attribute	Contents			
Depreciation	Line item depreciation withheld.			
Type	Format	Range	Required	Key
Currency	#.##		No	

INSURANCE_COS

Insurance companies.

INSURANCE_COS (InsuranceCo_ID)

Attribute	Contents			
#InsuranceCo_ID	Unique abbreviation key identifier, up to 7 characters.			
Type	Format	Range	Required	Key
Text			Yes	PK

INSURANCE_COS (Name)

Attribute	Contents			
Name	Company Name			
Type	Format	Range	Required	Key
Text			Yes	

INSURANCE_COS (Address)

Attribute	Contents			
Address	Street name and number of primary address.			
Type	Format	Range	Required	Key
Text			No	

INSURANCE_COS (ZipCode)

Attribute	Contents			
ZipCode (FK)	Zip Code associated with Insurance Company address			
Type	Format	Range	Required	Key
Text			No	FK ZIPS(ZipCode)

INSURANCE_COS (PhoneNumber)

Attribute	Contents			
PhoneNumber	7 digit primary phone number.			
Type	Format	Range	Required	Key
Text	###-####	000-0000 to 999-9999	No	

INSURANCE_COS (PhoneAreaCode)

Attribute	Contents			
PhoneAreaCode	Area code associated with primary phone number.			
Type	Format	Range	Required	Key
Text	###	000 to 999	No	

INSURANCE_COS (FaxNumber)

Attribute	Contents			
FaxNumber	7 digit primary fax number.			
Type	Format	Range	Required	Key
Text	###-####	000-0000 to 999-9999	No	

INSURANCE_COS (FaxAreaCode)

Attribute	Contents			
FaxAreaCode	Area code associated with primary fax number.			
Type	Format	Range	Required	Key
Text	###	000 to 999	No	

INSURANCE_COS (URL)

Attribute	Contents			
URL	Address of the insurance company's web site.			
Type	Format	Range	Required	Key
HYP			No	

INSURANCE_COS (Email)

Attribute	Contents			
Email	Insurance company's primary contact email address.			
Type	Format	Range	Required	Key
HYP			No	

INVOICES

Invoices received from vendors for materials and labor.

INVOICES (Invoice_ID)

Attribute	Contents			
#Invoice_ID	Numeric key value			
Type	Format	Range	Required	Key
AutoNumber (Long Integer)			Yes	PK

INVOICES (Vendor)

Attribute	Contents			
Vendor (FK)	Numeric identifier of vendor.			
Type	Format	Range	Required	Key
Number (Long Integer)			Yes	FK VENDORS (Vendor_ID)

INVOICES (Project)

Attribute	Contents			
Project (FK)	Numeric identifier of project.			
Type	Format	Range	Required	Key
Number (Long Integer)			No	FK PROJECTS (Project_ID)

INVOICES (InvoiceDate)

Attribute	Contents			
InvoiceDate	Date of invoice.			
Type	Format	Range	Required	Key
Date/Time			Yes	

ITEM_CATEGORIES

Categories of labor and material types. These codes and descriptions correspond to the Category List available from Xactimate, and may be imported directly from a data export from that program.

ITEM_CATEGORIES (ItemCode)*

Attribute	Contents			
Trade_ID	Unique abbreviation key value.			
Type	Format	Range	Required	Key
Text	###		Yes	PK

*The ItemCode directly corresponds to category codes available from Xactimate.

ITEM_CATEGORIES (Description)

Attribute	Contents			
Description	Trade name.			
Type	Format	Range	Required	Key
Text			Yes	

LABOR_TRADES

This table has been renamed to ITEM_CATEGORIES to better reflect the nature of its information and its role in overall business processes.

MATERIALS

Building materials estimated and purchased.

MATERIALS (Category)

Attribute	Contents			
#Category (FK)	3 character item category code corresponding to Xactimate codes.			
Type	Format	Range	Required	Key
Text	XXX		Yes	PK, FK ITEM_CATEGORIES (ItemCode)

MATERIALS (Material_ID)

Attribute	Contents			
#Material_ID	Material code used by Xactimate in conjunction with a category code to create a unique material identifier.			
Type	Format	Range	Required	Key
Text			Yes	PK

MATERIALS (Description)

Attribute	Contents			
Description	Material description			
Type	Format	Range	Required	Key
Text			Yes	

MATERIALS (Unit)*

Attribute	Contents			
Unit	Unit of measurement.			
Type	Format	Range	Required	Key
Text	XX	Value in: BF, CY, DA, EA, HR, LF, PL, SF, SQ, WK	No	

* To better map to Xactimate related processes and to reduce database management overhead, units have been moved from individual referencing tables and are now directly associated with the materials here.

PERMIT_CODES

Notes associated with permit requirements according to zip code.

PERMIT_CODES (Code_ID)

Attribute	Contents			
#Code_ID	Numeric key value.			
Type	Format	Range	Required	Key
AutoNumber (Long Integer)			Yes	PK

PERMIT_CODES (ZipCode)

Attribute	Contents			
ZipCode (FK)	Zip code for permit requirement.			
Type	Format	Range	Required	Key
Text	#####	00000-99999	Yes	FK ZIPS (ZipCode)

PERMIT_CODES (ContactName)

Attribute	Contents			
ContactName	Official to contact for information.			
Type	Format	Range	Required	Key
Text			No	

PERMIT_CODES (ContactPhoneNumber)

Attribute	Contents			
ContactPhoneNu mber	7 digit phone number of official contact.			
Type	Format	Range	Required	Key
Text	###-####	000-0000 to 999-9999	No	

PERMIT_CODES (ContactPhoneAreaCode)

Attribute	Contents			
ContactPhoneAreaCode	Area code associated with phone number of official contact.			
Type	Format	Range	Required	Key
Text	###	000 to 999	No	

PERMIT_CODES (Category)

Attribute	Contents			
Category (FK)	Character identifier of trade category associated with permit note.			
Type	Format	Range	Required	Key
Text	XXX		No	FK ITEM_CATEGOREIS (ItemCode)

PERMIT_CODES (PermitNote)

Attribute	Contents			
PermitNote	Description of permit requirements.			
Type	Format	Range	Required	Key
Text			Yes	

PHOTOS

System links to photographic image files associated with a project. This table utilizes the Access Hyperlink data type and requires systematic organization of the image files.

PHOTOS (Photo_ID)

Attribute	Contents			
#Photo_ID	Numeric key value.			
Type	Format	Range	Required	Key

AutoNumber (Long Integer)			Yes	PK
------------------------------	--	--	-----	----

PHOTOS (Project)

Attribute	Contents			
Project (FK)	Numeric project identifier.			
Type	Format	Range	Required	Key
Number (LongInteger)			Yes	FK PROJECTS (Project_ID)

PHOTOS (Description)

Attribute	Contents			
Description	Description of photographic image.			
Type	Format	Range	Required	Key
Text			No	

PHOTOS (Photo_DATE)

Attribute	Contents			
Photo_DATE	Date the photograph was taken.			
Type	Format	Range	Required	Key
Date/Time			No	

PHOTOS (Link_HYP)

Attribute	Contents			
Link_HYP	System link to photo image file.			
Type	Format	Range	Required	Key
HYP			Yes	

PRICELISTS

Standard lists of material and labor values. In the current business operations model, these lists are provided as a service from Xactimate, so key identifiers from this source will be used to identify individual price lists.

PRICELISTS (Pricelist_ID)

Attribute	Contents			
#Pricelist_ID	Identifying name of price list.			
Type	Format	Range	Required	Key
Text			Yes	PK

PRICELISTS (Quarter)

Attribute	Contents			
Quarter	Calendar quarter for which the pricelist was created.			
Type	Format	Range	Required	Key
Text	#	1 - 4	No	

PRICELISTS (PricelistYear)*

Attribute	Contents			
PricelistYear	Calendar year for which the pricelist was created.			
Type	Format	Range	Required	Key
Text	####	0000-9999	No	

*Because YEAR is a reserved word in SQL, this attribute has been changed to PricelistYear.

PRICELIST_PRICES

Line item values for materials and labor associated with a given pricelist.

PRICELISTS_PRICES (Pricelist)

Attribute	Contents			
#Pricelist (FK)	Unique pricelist name identifier.			
Type	Format	Range	Required	Key
Text			Yes	PK, FK PRICELISTS (Pricelist_ID)

PRICELISTS_PRICES (Category)

Attribute	Contents			
#Category	3 character category identifier.			
Type	Format	Range	Required	Key
Text	XXX		Yes	PK, FK MATERIALS (ItemCode)

PRICELISTS_PRICES (Material)

Attribute	Contents			
#Material (FK)	Character material identifier.			
Type	Format	Range	Required	Key
Text			Yes	PK, FK MATERIALS (Material_ID)

PRICELISTS_PRICES (UnitPrice)

Attribute	Contents			
UnitPrice	Line item price.			
Type	Format	Range	Required	Key
Currency	###		Yes	

PROJECTS

A remodeling project associated with a customer home and a date of incurred weather damage.

PROJECTS (Project_ID)

Attribute	Contents			
#Project_ID	Numeric key value.			
Type	Format	Range	Required	Key
AutoNumber (Long Integer)			Yes	PK

PROJECTS (Customer)

Attribute	Contents			
Customer (FK)	Numeric identifier of customer associated with project.			
Type	Format	Range	Required	Key
Number (Long Integer)			Yes	FK CUSTOMERS (Customer_ID)

PROJECTS (Address)

Attribute	Contents			
Address	Project address. May be same or differ from associated Customer address.			
Type	Format	Range	Required	Key
Text			No	

PROJECTS (ZipCode)

Attribute	Contents			
ZipCode	Zip Code associated with project address			
Type	Format	Range	Required	Key
Text	#####	00000-9999	No	FK ZIPS(ZipCode)

PROJECTS (Loss_DATE)

Attribute	Contents			
Loss_DATE	Date of loss associated with weather damage of home.			
Type	Format	Range	Required	Key
Date/Time			No	

PROJECTS (LossType)

Attribute	Contents			
LossType	Description of cause of damage.			
Type	Format	Range	Required	Key
Text			No	

PROJECTS (SalesRep)

Attribute	Contents			
SalesRep (FK)	Numeric identifier of employee responsible as sales contact to customer.			
Type	Format	Range	Required	Key
Number (Long Integer)			Yes	FK EMPLOYEES (Employee_ID)

PROJECTS (Status)

Attribute	Contents			
Status	Current status of the project in the project life cycle.			
Type	Format	Range	Required	Key

Text		Value in: Pending Sold In Progress Completed Awaiting Money Closed	No	
------	--	--	----	--

PROJECT_NOTES

Task, reminder and progress notes for tracking of regular communications and activities occurring in the project life cycle.

PROJECT_NOTES (Note_ID)

Attribute	Contents			
#Note_ID	Key value.			
Type	Format	Range	Required	Key
AutoNumber (Long Integer)			Yes	PK

PROJECT_NOTES (DATE_TIME)

Attribute	Contents			
DATE_TIME	Date and time note was recorded.			
Type	Format	Range	Required	Key
Date/Time			Yes	

PROJECT_NOTES (Project)

Attribute	Contents			
Project (FK)	Numeric identifier of project associated with note.			
Type	Format	Range	Required	Key

Number (Long Integer)			Yes	FK PROJECTS (Project_ID)
-----------------------	--	--	-----	--------------------------

PROJECT_NOTES (Note)

Attribute	Contents			
Note	Contents of note taken			
Type	Format	Range	Required	Key
Text			Yes	

PROJECT_NOTES (RecordedBy)

Attribute	Contents			
RecordedBy (FK)	Numeric identifier of employee who recorded the note.			
Type	Format	Range	Required	Key
Number (Long Integer)			No	FK EMPLOYEES (Employee_ID)

PURCHASES

Line item purchases associated with a vendor invoice.

PURCHASES (LineItem_ID)

Attribute	Contents			
#LineItem_ID (FK)	Numeric identifier of purchase item.			
Type	Format	Range	Required	Key
Number (Long Integer)			Yes	PK

PURCHASES (Invoice)

Attribute	Contents			
Invoice (FK)	Numeric identifier of invoice.			
Type	Format	Range	Required	Key
Number (Long Integer)			Yes	FK INVOICES (Invoice_ID)

PURCHASES (Category)

Attribute	Contents			
Category (FK)	3 character Xactimate category identifier.			
Type	Format	Range	Required	Key
Text	XXX		Yes	FK MATERIALS (Category)

PURCHASES (Material)

Attribute	Contents			
Material (FK)	Numeric identifier of material purchased			
Type	Format	Range	Required	Key
Number (Long Integer)			Yes	FK MATERIALS (Material_ID)

PURCHASES (Unit)

Attribute	Contents			
#Unit	Unit of measurement associated with line item.			
Type	Format	Range	Required	Key
Text		Value in: BF, CY, EA, LF, PL, SF, SQ	Yes	

PURCHASES (Quantity)

Attribute	Contents
Quantity	Quantity of material purchased

Type	Format	Range	Required	Key
Number	#.##		Yes	

PURCHASES (Cost)

Attribute	Contents			
Cost	Purchase price.			
Type	Format	Range	Required	Key
Currency			Yes	

PURCHASES (Style)

Attribute	Contents			
Style (FK)	Numeric identifier of material style purchased.			
Type	Format	Range	Required	Key
Number (Long Integer)			No	FK STYLES (Style_ID)

SCHEDULES

Tracking of project labor schedules.

SCHEDULES (Project)

Attribute	Contents			
#Project (FK)	Numeric identifier of project.			
Type	Format	Range	Required	Key
Number (Long Integer)			Yes	PK, FK PROJECTS (Project_ID)

SCHEDULES (Category)

Attribute	Contents			
-----------	----------	--	--	--

#Category (FK)	3 character Xactimate identifier of labor category.			
Type	Format	Range	Required	Key
Text	XXX		Yes	PK, FK LABOR_TRADES (Labor_ID)

SCHEDULES (Sub)

Attribute	Contents			
Sub (FK)	Numeric identifier of subcontractor.			
Type	Format	Range	Required	Key
Number (Long Integer)			No	FK SUBS (Sub_ID)

SCHEDULES (Status)

Attribute	Contents			
Status	Description of current scheduling status.			
Type	Format	Range	Required	Key
Text		Value in: Pending Scheduled In Progress Complete	No	

SCHEDULES (Workstart_DATE)

Attribute	Contents			
Workstart_DATE	Date labor is scheduled to begin.			
Type	Format	Range	Required	Key
Date/Time			No	

STATES

Static table of state names and their abbreviations.

STATES (Abbreviation)

Attribute	Contents			
#Abbreviation	Two character state abbreviation			
Type	Format	Range	Required	Key
Text	XX		Yes	PK

STATES (Name)

Attribute	Contents			
Name	Full state name			
Type	Format	Range	Required	Key
Text			Yes	

STYLES

Styles and colors for materials purchased. Particularly used to track selected colors and styles for roofing, siding and gutters.

STYLES (Style_ID)

Attribute	Contents			
#Style_ID	Numeric key value.			
Type	Format	Range	Required	Key
AutoNumber (LongInteger)			Yes	PK

STYLES (Description)

Attribute	Contents
-----------	----------

Description	Description of style and color.			
Type	Format	Range	Required	Key
Text			Yes	

STYLES (Material)

Attribute	Contents			
Material (FK)	Numeric identifier of material associated with style.			
Type	Format	Range	Required	Key
Number (Long Integer)			Yes	FK MATERIALS (Material_ID)

SUBS

Subcontractors hired to perform remodeling labor.

SUBS (Sub_ID)

Attribute	Contents			
#Sub_ID	Numeric key value.			
Type	Format	Range	Required	Key
AutoNumber (Long Integer)			Yes	PK

SUBS (Category)

Attribute	Contents			
Category (FK)	Primary Xactimate category associated with sub-contractor's specialization			
Type	Format	Range	Required	Key
Text	XXX		No	FK ITEM_CATEGORIES (ItemCode)

SUBS (Name)

Attribute	Contents			
Name	Name of contracting independent or company.			
Type	Format	Range	Required	Key
Text			Yes	

SUBS (Address)

Attribute	Contents			
Address	Street name and number of subcontractor.			
Type	Format	Range	Required	Key
Text			No	

SUBS (ZipCode)

Attribute	Contents			
ZipCode (FK)	Zip Code associated with subcontractor address.			
Type	Format	Range	Required	Key
Text	#####	00000-99999	Yes	FK ZIPS(ZipCode)

SUBS (PhoneNumber)

Attribute	Contents			
PhoneNumber	Subcontractor's 7 digit phone number.			
Type	Format	Range	Required	Key
Text	###-####	000-0000 to 999-9999	No	

SUBS (PhoneAreaCode)

Attribute	Contents			
PhoneAreaCode	Area code associated with subcontractor's phone number.			
Type	Format	Range	Required	Key

Text	###	000 to 999	No	
------	-----	------------	----	--

SUBS (ContactName)

Attribute	Contents			
ContactName	Contact name if different from primary name.			
Type	Format	Range	Required	Key
Text			No	

SUBS (ContactPhoneNumber)

Attribute	Contents			
ContactPhoneNu mber	7 digit telephone number of contact if different from primary phone number.			
Type	Format	Range	Required	Key
Text	###-####	000-0000 to 999-9999	No	

SUBS (ContactAreaCode)

Attribute	Contents			
ContactAreaCode	Area code associated with contact's phone number			
Type	Format	Range	Required	Key
Text	###	000 to 999	No	

VENDORS

Materials suppliers.

VENDORS (Vendor_ID)

Attribute	Contents			
#Vendor_ID	Numeric key value.			
Type	Format	Range	Required	Key
AutoNumber (Long Integer)			Yes	PK

VENDORS (Name)

Attribute	Contents			
Name	Name of vending company			
Type	Format	Range	Required	Key
Text			Yes	

VENDORS (Address)

Attribute	Contents			
Address	Street name and number of vendor's address.			
Type	Format	Range	Required	Key
Text			No	

VENDORS (ZipCode)

Attribute	Contents			
ZipCode (FK)	Zip Code associated with vendor address.			
Type	Format	Range	Required	Key
Text	#####	00000-99999	Yes	FK ZIPS(ZipCode)

VENDORS (PhoneNumber)

Attribute	Contents			
PhoneNumber	Vendor's primary 7 digit phone number.			
Type	Format	Range	Required	Key
Text	###-####	000-0000 to 999-9999	No	

VENDORS (PhoneAreaCode)

Attribute	Contents			
PhoneAreaCode	Area code associated with vendor's primary phone number.			
Type	Format	Range	Required	Key
Text	###	000 to 999	No	

VENDORS (Email)

Attribute	Contents			
Email	Vendor's email address			
Type	Format	Range	Required	Key
HYP			No	

VENDORS (ContactName)

Attribute	Contents			
ContactName	Primary contact person at vendor.			
Type	Format	Range	Required	Key
Text			No	

VENDORS (ContactPhoneNumber)

Attribute	Contents			
ContactPhoneNu mber	Primary contact's 7 digit phone number if different from primary business number.			
Type	Format	Range	Required	Key
Text	###-####	000-0000 to 999-9999	No	

VENDORS (ContactPhoneAreaCode)

Attribute	Contents			
-----------	----------	--	--	--

ContactPhoneAreaCode	Area code associated with primary contact's phone number.			
Type	Format	Range	Required	Key
Text	###	000 to 999	No	

ZIPs

Zip codes providing key references to cities and states. The current business model assumes a limited geographic business region, thus Zip codes will be used to ensure address integrity in a number of contact tables and to reduce the amount of data stored. Plus 4 codes will not be stored in this table, but in the local table with other address information.

ZIPs (ZipCode)

Attribute	Contents			
#ZipCode	5 digit postal code.			
Type	Format	Range	Required	Key
Text	#####	00000 to 99999	Yes	PK

ZIPs (City)

Attribute	Contents			
City	City name associated with zip code.			
Type	Format	Range	Required	Key
Text			Yes	

ZIPs (State)

Attribute	Contents			
State (FK)	State abbreviation associated with city and zip code.			
Type	Format	Range	Required	Key

Text	XX		Yes	FK STATES (Abbreviation)
------	----	--	-----	-----------------------------

SQL Code

The following code demonstrates the building of the designed database with SQL. In access, each SQL should be executed separately as Access only allows for a single SQL statement to be executed at a time.

Table creation should occur in the order demonstrated by the following code, since that order has been arranged to account for the creation of foreign entities before the creation of related foreign constraints.

Additionally, several manual changes must be made as noted below. These additional settings are necessary in order to adhere to the design as developed through this document. The need for manual creation of these settings occurs because Access treats several application level constraints as if they were database level data types. Such “data type” settings may be made in the design view of the respective table. It is imperative that these settings be made immediately after creating the given table, before other tables are created, since with the creation of later tables, relationships are also created which in turn prevent the data types from being further modified.

The indicated SQL table creation code and necessary manual table modifications follow.

Table Name: STATES

Primary Key: Abbreviation

Foreign Key(s): n/a

SQL table creation:

```
CREATE TABLE STATES (
    Abbreviation TEXT(2) NOT NULL,
    CONSTRAINT states_pk PRIMARY KEY (Abbreviation),
```

```
Name TEXT(11) NOT NULL
```

```
);
```

Table Name: ZIPS

Primary Key: ZipCode

Foreign Key(s): STATES(Abbreviation)

SQL table creation:

```
CREATE TABLE ZIPS (  
  
    ZipCode TEXT(5) NOT NULL,  
    CHECK (ZipCode BETWEEN '00000' AND '99999'),  
    CONSTRAINT zips_pk PRIMARY KEY (ZipCode),  
  
    City TEXT(20) NOT NULL,  
  
    State TEXT(2) NOT NULL,  
    CONSTRAINT zips_state_fk FOREIGN KEY (State) REFERENCES  
    STATES(Abbreviation)  
  
);
```

Table Name: ADJUSTORS

Primary Key: Adjustor_ID

Foreign Key(s): ZIPS(ZipCode)

SQL table creation:

```
CREATE TABLE ADJUSTORS (  
  
    Adjustor_ID INTEGER NOT NULL,  
    CONSTRAINT adjustors_pk PRIMARY KEY (Adjustor_ID),  
  
    FirstName TEXT(20),
```

```
LastName TEXT(20),  
Address TEXT(50),  
  
ZipCode TEXT(5),  
CONSTRAINT adjustors_zipcode_fk FOREIGN KEY (ZipCode) REFERENCES  
ZIPS(ZipCode),  
  
PhoneAreaCode TEXT(3),  
CHECK (PhoneAreaCode BETWEEN '000' AND '999'),  
  
PhoneNumber TEXT(7),  
CHECK (PhoneNumber BETWEEN '0000000' AND '9999999'),  
  
FaxAreaCode TEXT(3),  
CHECK (FaxAreaCode BETWEEN '000' AND '999'),  
  
FaxNumber TEXT(7),  
CHECK (FaxNumber BETWEEN '0000000' AND '9999999'),  
  
Email TEXT(30)  
);
```

Important manual configuration:

Set ADJUSTORS (Adjustor_ID) to the AutoNumber type.

Set ADJUSTORS (Email) to the Hyperlink type.

Table Name: EMPLOYEES

Primary Key: Employee_ID

Foreign Key(s): ZIPS(ZipCode)

SQL table creation:

```
CREATE TABLE EMPLOYEES (
```

```
Employee_ID INTEGER NOT NULL,  
CONSTRAINT employees_pk PRIMARY KEY (Employee_ID),  
  
FirstName TEXT(20) NOT NULL,  
LastName TEXT(20) NOT NULL,  
Address TEXT(50),  
  
ZipCode TEXT(5),  
CONSTRAINT employees_zipcode_fk FOREIGN KEY (ZipCode)  
REFERENCES ZIPS(ZipCode),  
  
PhoneNumber TEXT(7),  
CHECK (PhoneNumber BETWEEN '0000000' AND '9999999'),  
  
PhoneAreaCode TEXT(3),  
CHECK (PhoneAreaCode BETWEEN '000' AND '999'),  
  
Role TEXT(30)  
  
);
```

Important manual configuration:

Set EMPLOYEES (Employee_ID) to the AutoNumber type.

Table Name: INSURANCE_COS

Primary Key: InsuranceCo_ID

Foreign Key(s): ZIPS(ZipCode)

SQL table creation:

```
CREATE TABLE INSURANCE_COS (  
  
InsuranceCo_ID TEXT(7) NOT NULL,  
CONSTRAINT insurance_cos_pk PRIMARY KEY (InsuranceCo_ID),
```

```
Name TEXT(30) NOT NULL,  
Address TEXT(50),  
  
ZipCode TEXT(5),  
CONSTRAINT insurance_cos_zipcode_fk FOREIGN KEY (ZipCode)  
REFERENCES ZIPS(ZipCode),  
  
PhoneNumber TEXT(7),  
CHECK (PhoneNumber BETWEEN '0000000' AND '9999999'),  
  
PhoneAreaCode TEXT(3),  
CHECK (PhoneAreaCode BETWEEN '000' AND '999'),  
  
FaxNumber TEXT(7),  
CHECK (FaxNumber BETWEEN '0000000' AND '9999999'),  
  
FaxAreaCode TEXT(3),  
CHECK (FaxAreaCode BETWEEN '000' AND '999'),  
  
URL TEXT(50),  
  
Email TEXT(50)  
  
);
```

Important manual configuration:

Set INSURANCE_COS (URL) to the Hyperlink type.

Set INSURANCE_COS (Email) to the Hyperlink type.

Table Name: CUSTOMERS

Primary Key: Customer_ID

Foreign Key(s): ZIPS(ZipCode), INSURANCE_COS(InsuranceCo_ID)

SQL table creation:

```
CREATE TABLE CUSTOMERS (  
  
Customer_ID INTEGER NOT NULL,  
CONSTRAINT customers_pk PRIMARY KEY (Customer_ID),  
  
FirstName TEXT(20) NOT NULL,  
LastName TEXT(20) NOT NULL,  
Salutation TEXT(50),  
Address TEXT(50) NOT NULL,  
  
ZipCode TEXT(5),  
CONSTRAINT customers_zipcode_fk FOREIGN KEY (ZipCode)  
REFERENCES ZIPS(ZipCode),  
  
InsuranceCo TEXT(7),  
CONSTRAINT customers_insuranceco_fk FOREIGN KEY (InsuranceCo)  
REFERENCES INSURANCE_COS(InsuranceCo_ID),  
  
PolicyNo TEXT(20),  
  
PhoneNumber TEXT(7),  
CHECK (PhoneNumber BETWEEN '0000000' AND '9999999'),  
  
PhoneAreaCode TEXT(3),  
CHECK (PhoneNumber BETWEEN '000' AND '999')  
);
```

Important manual configuration:

Set CUSTOMERS (Customer_ID) to the AutoNumber type.

Table Name: PROJECTS

Primary Key: Project_ID

**Foreign Key(s): CUSTOMERS(Customer_ID), EMPLOYEES(Employee_ID),
ZIPS(ZipCode)**

SQL table creation:

```
CREATE TABLE PROJECTS (  
  
Project_ID INTEGER NOT NULL,  
CONSTRAINT projects_pk PRIMARY KEY (Project_ID),  
  
Customer INTEGER NOT NULL,  
CONSTRAINT projects_customer_fk FOREIGN KEY (Customer)  
REFERENCES CUSTOMERS(Customer_ID),  
  
Address TEXT(50),  
  
ZipCode TEXT(5),  
CONSTRAINT projects_zipcode_fk FOREIGN KEY (ZipCode) REFERENCES  
ZIPS(ZipCode),  
  
Loss_DATE DATE,  
LossType TEXT(10),  
  
SalesRep INTEGER NOT NULL,  
CONSTRAINT projects_salesrep_fk FOREIGN KEY (SalesRep) REFERENCES  
EMPLOYEES(Employee_ID),  
  
Status TEXT(14),  
CHECK (Status IN ('Pending','Sold', 'In Progress', 'Completed', 'Awaiting Money',  
'Closed'))  
  
);
```

Important manual configuration:

Set PROJECTS (Project_ID) to the AutoNumber type.

Table Name: ITEM_CATEGORIES

Primary Key: ItemCode

Foreign Key(s): n/a

SQL table creation:

```
CREATE TABLE ITEM_CATEGORIES (  
  
ItemCode TEXT(3) NOT NULL,  
CONSTRAINT item_categories PRIMARY KEY (ItemCode),  
  
Description TEXT(30) NOT NULL  
  
);
```

Table Name: MATERIALS

Primary Key: Material_ID

Foreign Key(s): ITEM_CATEGORIES(ItemCode)

SQL table creation:

```
CREATE TABLE MATERIALS (  
  
Category TEXT(3) NOT NULL,  
CONSTRAINT materials_category_fk FOREIGN KEY (Category) REFERENCES  
ITEM_CATEGORIES(ItemCode),  
  
Material_ID TEXT(7) NOT NULL,  
  
CONSTRAINT materials_pk PRIMARY KEY (Category,Material_ID),  
  
Description TEXT(60) NOT NULL,  
  
Unit TEXT(20),  
CHECK (Unit IN('BF','CY','DA','EA','HR','LF','PL','SF','SQ','WK'))  
  
);
```

Table Name: PRICELISTS

Primary Key: Pricelist_ID

Foreign Key(s): ITEM_CATEGORIES(ItemCode)

SQL table creation:

```
CREATE TABLE PRICELISTS (  
  
Pricelist_ID Text(10) NOT NULL,  
CONSTRAINT pricelists_pk PRIMARY KEY (Pricelist_ID),  
  
Quarter Text(1),  
CHECK (Quarter BETWEEN '1' AND '4'),  
  
PricelistYear Text(4),  
CHECK (PricelistYear BETWEEN '0000' AND '9999')  
  
);
```

Table Name: PRICELIST_PRICES

Primary Key: Pricelist, Category, Material

Foreign Key(s): PRICELISTS(Pricelist_ID), MATERIALS(Category,Material_ID)

SQL table creation:

```
CREATE TABLE PRICELIST_PRICES (  
  
Pricelist TEXT(10) NOT NULL,  
CONSTRAINT pricelist_prices_pricelist_fk FOREIGN KEY (Pricelist)  
REFERENCES PRICELISTS(Pricelist_ID),  
  
Category TEXT(3) NOT NULL,  
Material TEXT(7) NOT NULL,
```

```
CONSTRAINT pricelist_prices_material_fk FOREIGN KEY (Category,Material)
REFERENCES MATERIALS(Category,Material_ID),

CONSTRAINT pricelist_prices_pk PRIMARY KEY (Pricelist,Category,Material),

UnitPrice CURRENCY

);
```

Table Name: BIDS

Primary Key: Bid_ID

**Foreign Key(s): PROJECTS(Project_ID), PRICELISTS(Pricelist_ID),
EMPLOYEES(Employee_ID)**

SQL table creation:

```
CREATE TABLE BIDS (

Bid_ID INTEGER NOT NULL,
CONSTRAINT bids_pk PRIMARY KEY (Bid_ID),

Project INTEGER NOT NULL,
CONSTRAINT bids_project_fk FOREIGN KEY (Project) REFERENCES
PROJECTS(Project_ID),

Bid_DATE DATE NOT NULL,

Pricelist TEXT(10),
CONSTRAINT bids_pricelist_fk FOREIGN KEY (Pricelist) REFERENCES
PRICELISTS(Pricelist_ID),

Estimator INTEGER,
CONSTRAINT bids_estimator_fk FOREIGN KEY (Estimator) REFERENCES
EMPLOYEES(Employee_ID),
```

```
Inspected_DATE DATE
```

```
);
```

Important manual configuration:

Set BIDS (Bid_ID) to the AutoNumber type.

Table Name: BID_PRICES

Primary Key: Bid, Category, Material

Foreign Key(s): BIDS(Bid_ID), MATERIALS(Category,Material_ID)

EMPLOYEES(Employee_ID)

SQL table creation:

```
CREATE TABLE BID_PRICES (  
  
    Bid INTEGER NOT NULL,  
    CONSTRAINT bid_prices_bid_fk FOREIGN KEY (Bid)  
    REFERENCES BIDS(Bid_ID),  
  
    Category TEXT(3) NOT NULL,  
    Material TEXT(7) NOT NULL,  
  
    CONSTRAINT bid_prices_material_fk FOREIGN KEY (Category,Material)  
    REFERENCES MATERIALS(Category,Material_ID),  
  
    CONSTRAINT bid_prices_pk PRIMARY KEY (Bid,Category,Material),  
  
    UnitPrice CURRENCY,  
    Quantity LONG  
  
);
```

Table Name: ESTIMATES

Primary Key: Estimate_ID

**Foreign Key(s): CUSTOMERS(Customer_ID),
INSURANCE_COS(InsuranceCo_ID), ADJUSTORS(Adjustor_ID),
PRICELISTS(Pricelist_ID)**

SQL table creation:

```
CREATE TABLE ESTIMATES (  
  
    Estimate_ID INTEGER NOT NULL,  
    CONSTRAINT estimates_pk PRIMARY KEY (Estimate_ID),  
  
    Customer INTEGER NOT NULL,  
    CONSTRAINT estimates_customer_fk FOREIGN KEY (Customer)  
    REFERENCES CUSTOMERS(Customer_ID),  
  
    Loss_DATE DATE,  
    Estimate_DATE DATE,  
    Inspected_DATE DATE,  
  
    Insurance TEXT(7),  
    CONSTRAINT estimates_insurance_fk FOREIGN KEY (Insurance)  
    REFERENCES INSURANCE_COS(InsuranceCo_ID),  
  
    Adjustor INTEGER,  
    CONSTRAINT estimates_adjustor FOREIGN KEY (Adjustor) REFERENCES  
    ADJUSTORS(Adjustor_ID),  
  
    SalesTax LONG,  
    CHECK (SalesTax BETWEEN 0.00 AND 0.99),  
  
    Depreciation CURRENCY,  
    TotalClaim CURRENCY,  
  
    Pricelist TEXT(10),  
    PolicyNo TEXT(20),  
    ClaimNo TEXT(20),
```

```
ACV CURRENCY,  
PWI CURRENCY,  
Deductible CURRENCY,  
OandP CURRENCY,  
DepreciationStatus TEXT(10),  
PWI_Status TEXT(10),  
OandP_Status TEXT(10)  
);
```

Important manual configuration:

Set ESTIMATES (Estimate_ID) to the AutoNumber type.

Table Name: ESTIMATE_PRICES

Primary Key: Estimate, Category, Material, Operation

Foreign Key(s): ESTIMATES(Estimate_ID), MATERIALS(Category,Material_ID)

SQL table creation:

```
CREATE TABLE ESTIMATE_PRICES (  
  
Estimate INTEGER NOT NULL,  
CONSTRAINT estimate_prices_estimate FOREIGN KEY (Estimate)  
REFERENCES ESTIMATES(Estimate_ID),  
  
Category TEXT(3) NOT NULL,  
Material TEXT(7) NOT NULL,  
  
CONSTRAINT estimate_prices_material_fk FOREIGN KEY (Category,Material)  
REFERENCES MATERIALS(Category,Material_ID),  
  
Operation TEXT(7) NOT NULL,  
CHECK (Operation IN ('REMOVE','REPLACE','R&R','DET&RST')),
```

```
CONSTRAINT estimate_prices_pk PRIMARY KEY  
(Estimate,Category,Material,Operation),  
  
UnitPrice CURRENCY,  
Quantity LONG,  
Depreciation CURRENCY  
  
);
```

Important manual configuration:

Set ESTIMATE_PRICES (Estimate) to the AutoNumber type.

Table Name: VENDORS

Primary Key: Vendor_ID

Foreign Key(s): ZIPS(ZipCode)

SQL table creation:

```
CREATE TABLE VENDORS (  
  
Vendor_ID INTEGER NOT NULL,  
CONSTRAINT vendors_pk PRIMARY KEY (Vendor_ID),  
  
Name Text(30) NOT NULL,  
Address Text(50),  
ZipCode TEXT(5),  
CONSTRAINT vendors_zipcode_fk FOREIGN KEY (ZipCode) REFERENCES  
ZIPS(ZipCode),  
PhoneNumber Text(7),  
CHECK (PhoneNumber BETWEEN '0000000' AND '9999999'),  
  
PhoneAreaCode Text(3),  
CHECK (PhoneAreaCode BETWEEN '000' AND '999'),  
)
```

```
Email Text(50),  
  
ContactName Text(40),  
  
ContactPhoneNumber Text(7),  
CHECK (ContactPhoneNumber BETWEEN '0000000' AND '9999999'),  
  
ContactAreaCode Text(3),  
CHECK (ContactAreaCode BETWEEN '000' AND '999')  
);
```

Important manual configuration:

Set VENDORS (Vendor_ID) to the AutoNumber type.

Set VENDORS (Email) to the Hyperlink type.

Table Name: INVOICES

Primary Key: Invoice_ID

Foreign Key(s): VENDORS(Vendor_ID), PROJECTS(Project_ID)

SQL table creation:

```
CREATE TABLE INVOICES (  
  
Invoice_ID INTEGER NOT NULL,  
CONSTRAINT invoices_pk PRIMARY KEY (Invoice_ID),  
  
Vendor INTEGER NOT NULL,  
CONSTRAINT invoices_vendor_fk FOREIGN KEY (Vendor) REFERENCES  
VENDORS(Vendor_ID),  
  
Project INTEGER,  
CONSTRAINT invoices_project_fk FOREIGN KEY (Project) REFERENCES  
PROJECTS(Project_ID),
```

```
InvoiceDate DATE NOT NULL
```

```
);
```

Important manual configuration:

Set INVOICES (Invoice_ID) to the AutoNumber type.

Table Name: PERMIT_CODES

Primary Key: Code_ID

Foreign Key(s): ZIPS(ZipCode), ITEM_CATEGORIES(ItemCode)

SQL table creation:

```
CREATE TABLE PERMIT_CODES (  
  
Code_ID INTEGER NOT NULL,  
CONSTRAINT permit_codes_pk PRIMARY KEY (Code_ID),  
  
ZipCode TEXT(5),  
CONSTRAINT permit_codes_zipcode_fk FOREIGN KEY (ZipCode)  
REFERENCES ZIPS(ZipCode),  
  
ContactName Text(40),  
  
ContactPhoneNumber Text(7),  
CHECK (ContactPhoneNumber BETWEEN '0000000' AND '9999999'),  
  
ContactPhoneAreaCode Text(3),  
CHECK (ContactPhoneAreaCode BETWEEN '000' AND '999'),  
  
Category TEXT(3),  
CONSTRAINT permit_codes_category_fk FOREIGN KEY (Category)  
REFERENCES ITEM_CATEGORIES(ItemCode),  
  
PermitNote TEXT(100)
```

```
);
```

Important manual configuration:

Set PERMIT_CODES (Code_ID) to the AutoNumber type.

Table Name: PHOTOS

Primary Key: Photo_ID

Foreign Key(s): PROJECTS(Project_ID)

SQL table creation:

```
CREATE TABLE PHOTOS (  
  
    Photo_ID INTEGER NOT NULL,  
    CONSTRAINT photos_pk PRIMARY KEY (Photo_ID),  
  
    Project INTEGER NOT NULL,  
    CONSTRAINT photos_project_fk FOREIGN KEY (Project) REFERENCES  
    PROJECTS(Project_ID),  
  
    Description TEXT(50),  
    Photo_DATE DATE,  
    Link_HYP TEXT(20) NOT NULL  
  
);
```

Important manual configuration:

Set PHOTOS (Photo_ID) to the AutoNumber type.

Set PHOTOS (Link) to the Hyperlink type. Set PROJECT_NOTES
(Note_ID) to the AutoNumber type.

Table Name: PROJECT_NOTES

Primary Key: Note_ID

Foreign Key(s): PROJECTS(Project_ID), EMPLOYEES(Employee_ID)

SQL table creation:

```
CREATE TABLE PROJECT_NOTES (  
  
    Note_ID INTEGER NOT NULL,  
    CONSTRAINT project_notes_pk PRIMARY KEY (Note_ID),  
  
    DATE_TIME TIMESTAMP NOT NULL,  
  
    Project INTEGER NOT NULL,  
    CONSTRAINT project_notes_project_fk FOREIGN KEY (Project)  
    REFERENCES PROJECTS(Project_ID),  
  
    NoteContent TEXT(200) NOT NULL,  
  
    RecordedBy INTEGER NOT NULL,  
    CONSTRAINT project_notes_recordedby_fk FOREIGN KEY (RecordedBy)  
    REFERENCES EMPLOYEES(Employee_ID)  
  
);
```

Table Name: STYLES

Primary Key: Style_ID

Foreign Key(s): MATERIALS(Category,Material_ID)

SQL table creation:

```
CREATE TABLE STYLES (  
  
    Style_ID TEXT(10) NOT NULL,  
    CONSTRAINT styles_pk PRIMARY KEY (Style_ID),  
  
    Description TEXT(50),  
  
    Category TEXT(3),
```

```
Material TEXT(7),  
CONSTRAINT styles_material_fk FOREIGN KEY (Category,Material)  
REFERENCES MATERIALS(Category,Material_ID)  
  
);
```

Important manual configuration:

Set STYLES (Style_ID) to the AutoNumber type.

Table Name: PURCHASES

Primary Key: LineItem_ID

**Foreign Key(s): INVOICES(Invoice_ID), MATERIALS(Category,Material_ID),
STYLES(Style_ID)**

SQL table creation:

```
CREATE TABLE PURCHASES (  
  
LineItem_ID INTEGER NOT NULL,  
CONSTRAINT purchases_pk PRIMARY KEY (LineItem_ID),  
  
Invoice INTEGER NOT NULL,  
CONSTRAINT purchases_invoice_fk FOREIGN KEY (Invoice) REFERENCES  
INVOICES(Invoice_ID),  
  
Category TEXT(3) NOT NULL,  
Material TEXT(7) NOT NULL,  
CONSTRAINT purchases_material_fk FOREIGN KEY (Category,Material)  
REFERENCES MATERIALS(Category,Material_ID),  
  
Unit TEXT(2) NOT NULL,  
CHECK (Unit IN('BF','CY','EA','LF','PL','SF','SQ')),  
  
Quantity LONG NOT NULL,  
  
Cost CURRENCY NOT NULL,
```

```
Style TEXT(10),  
CONSTRAINT purchases_style_fk FOREIGN KEY (Style) REFERENCES  
STYLES(Style_ID)  
);
```

Important manual configuration:

Set PURCHASES (LineItem_ID) to the AutoNumber type.

Table Name: SUBS

Primary Key: Sub_ID

Foreign Key(s): ITEM_CATEGORIES(ItemCode), ZIPS(ZipCode)

SQL table creation:

```
CREATE TABLE SUBS (  
  
Sub_ID INTEGER NOT NULL,  
CONSTRAINT subs_pk PRIMARY KEY (Sub_ID),  
  
Category TEXT(3),  
CONSTRAINT subs_category_fk FOREIGN KEY (Category) REFERENCES  
ITEM_CATEGORIES(ItemCode),  
Name TEXT(20),  
  
Address TEXT(50),  
  
ZipCode TEXT(5),  
CONSTRAINT subs_zipcode_fk FOREIGN KEY (ZipCode) REFERENCES  
ZIPS(ZipCode),  
  
PhoneNumber Text,  
CHECK (PhoneNumber BETWEEN '0000000' AND '9999999'),  
  
PhoneAreaCode Text,
```

```
CHECK (PhoneAreaCode BETWEEN '000' AND '999'),

ContactName Text(40),

ContactPhoneNumber Text,
CHECK (ContactPhoneNumber BETWEEN '0000000' AND '9999999'),

ContactAreaCode Text,
CHECK (ContactAreaCode BETWEEN '000' AND '999')

);
```

Important manual configuration:

Set SUBS (Sub_ID) to the AutoNumber type.

Table Name: SCHEDULES

Primary Key: Project, Category

**Foreign Key(s): PROJECTS(Project_ID), ITEM_CATEGORIES(ItemCode),
SUBS(Sub_ID)**

SQL table creation:

```
CREATE TABLE SCHEDULES (

Project INTEGER NOT NULL,
CONSTRAINT schedules_project_fk FOREIGN KEY (Project) REFERENCES
PROJECTS (Project_ID),

Category TEXT(3) NOT NULL,
CONSTRAINT schedules_category_fk FOREIGN KEY (Category)
REFERENCES ITEM_CATEGORIES(ItemCode),

CONSTRAINT schedules_pk PRIMARY KEY (Project,Category),

Sub INTEGER,
```

```
CONSTRAINT schedules_sub_fk FOREIGN KEY (Sub) REFERENCES  
SUBS(Sub_ID),  
  
Status TEXT(11),  
CHECK (Status IN ('Pending','Scheduled','In Progress','Complete')),  
  
Workstart_DATE DATE  
  
);
```

Forms Design

The complete design of the front-end database application is beyond the scope of this engineering document. However, MS Access allows us to quickly build data entry forms based on existing tables, thus allowing for the rapid creation of a front-end prototype. This section will focus on the potential for rapid front-end forms development for the General Contractor Database.

In order to create a form based on an existing table, the following steps are required:

- From the Forms window select “Create form by using wizard”.
- In the wizard, select the desired table and fields. In most cases, for our purposes of a rapid prototype, all fields except for artificial key values should be chosen. Key values that demonstrate interpretable information may also be selected (such as the Insurance abbreviation key, or the State 2 character abbreviation).
- It is important to select all NOT NULL columns, unless additional application provisions have been made to fill this data (such as the use of AutoNumber for key values)

In order to demonstrate the potential for rapid development of a user interface, the sub-system of project creation has been developed. This sub-system requires tracking Sales Representatives, Customers, Insurance Companies, and the projects that arise from the relationships of these entities. The following assumptions will be made for the purposes of this demonstration:

- These tables will be needed:

CUSTOMERS

PROJECTS

INSURANCE_COS

EMPLOYEES

ZIPS

STATES

- Since ZIPS and STATES are relatively static tables, these will be assumed to be pre-filled by a database administrator. The tables below show likely pre-filled data.

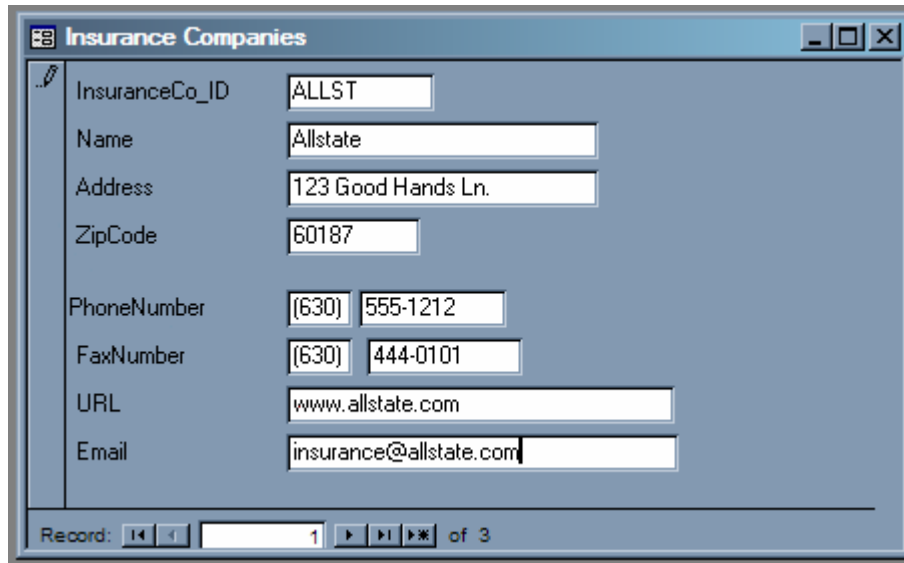
Abbreviation	Name
IL	Illinois
IN	Indiana
WI	Wisconsin
MI	Michigan

Table 4. Pre-filled STATES data

ZipCode	City	State
60187	Wheaton	IL
60007	Elk Grove Village	IL
60018	Rosemont	IL
60025	Glenview	IL
60030	Grayslake	IL
60072	Ringwood	IL
60076	Skokie	IL
60094	Paltine	IL
60101	Addison	IL
60103	Hanover Park	IL
60107	Streamwood	IL
60108	Bloomington	IL
60133	Hanover Park	IL
60134	Geneva	IL
60137	Glen Ellyn	IL

Table 5. Pre-filled ZIPS data

Figures 28-30 show layouts for the rapidly developed forms for the tables INSURANCE_COS, CUSTOMERS, and EMPLOYEES.

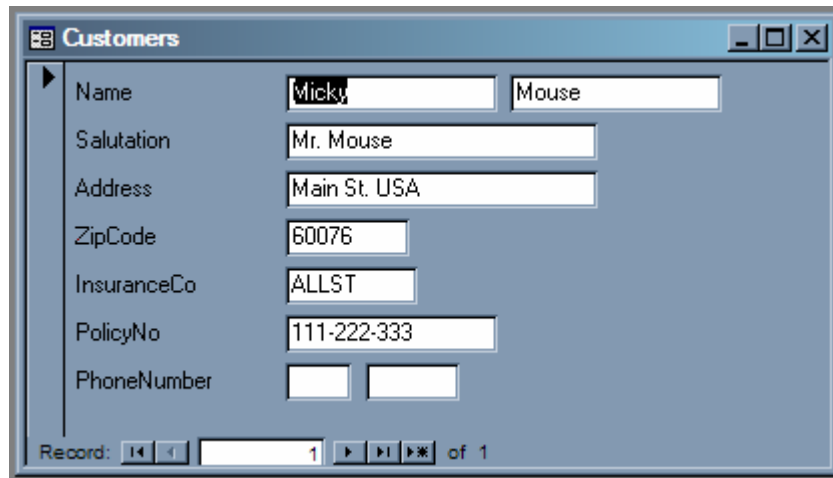


The 'Insurance Companies' form contains the following fields and values:

Field	Value
InsuranceCo_ID	ALLST
Name	Allstate
Address	123 Good Hands Ln.
ZipCode	60187
PhoneNumber	(630) 555-1212
FaxNumber	(630) 444-0101
URL	www.allstate.com
Email	insurance@allstate.com

Record: 1 of 3

Figure 28. Insurance Companies Data Entry Form

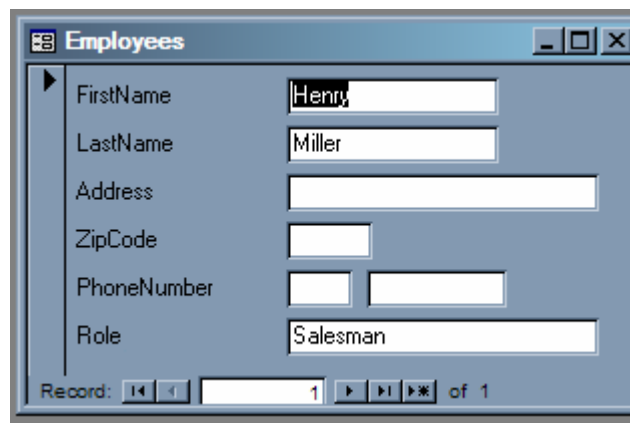


The 'Customers' form contains the following fields and values:

Field	Value
Name	Micky Mouse
Salutation	Mr. Mouse
Address	Main St. USA
ZipCode	60076
InsuranceCo	ALLST
PolicyNo	111-222-333
PhoneNumber	

Record: 1 of 1

Figure 29. Customers Data Entry Form



The 'Employees' form contains the following fields and values:

Field	Value
FirstName	Henw
LastName	Miller
Address	
ZipCode	
PhoneNumber	
Role	Salesman

Record: 1 of 1

Figure 30. Employees Data Entry Form.

Bringing this information together, Figure ____ shows the rapidly developed Projects view which includes data from the CUSTOMERS, EMPLOYEES and PROJECTS tables. This form is based on a query from those three tables. Being made up of a query of several tables, this form is not updatable, but is simply a view into the available data. For the completion of the Project sub-system, a Projects data-entry form may be needed, incorporating the use of sub-forms as necessary in order to accommodate the inclusion of several tables.

Figure 31. Projects Quick View Form.

This section has briefly introduced the potential for rapidly developing forms-based user interfaces to the General Contractor Database. User feedback, and standard development procedures would then provide the guidance needed for completing a more polished interface to the system. As this engineering document is meant to focus on the database itself, we will leave our discussion of User interface with this development potential in mind.

Turning our focus back to the database level, we will now demonstrate some queries which will be useful for turning the relational database into a true business information system. The following section contains data queries written in SQL for the purpose of gathering useful information from the database.

SQL Query Examples

Find all customer names, and the respective insurance company IDs with a project date-of-loss as March 8, 2004:

```
SELECT CUSTOMERS.FirstName, CUSTOMERS.LastName, InsuranceCo,
Loss_DATE FROM CUSTOMERS, PROJECTS
WHERE Loss_DATE = #3/8/04#
AND
```

```
CUSTOMERS.Customer_ID = PROJECTS.Customer
```

Find all project notes with associated Project_ID and employee who took the note for the date March 8, 2004.

```
SELECT Project, DATE_TIME, NoteContent, EMPLOYEES.FirstName,  
EMPLOYEES.LastName FROM PROJECT_NOTES, EMPLOYEES  
WHERE  
DATE_TIME LIKE #3/8/04#  
AND  
RecordedBy=Employee_ID
```

Find the same notes as above, but only those notes taken by the employee named Joe

```
SELECT Project, DATE_TIME, NoteContent, EMPLOYEES.FirstName,  
EMPLOYEES.LastName FROM PROJECT_NOTES, EMPLOYEES  
WHERE  
DATE_TIME LIKE #3/8/04#  
AND  
RecordedBy=Employee_ID  
AND  
FirstName="Joe"
```

Find all purchases of white siding

```
SELECT PURCHASES.* FROM PURCHASES, STYLES  
WHERE  
PURCHASES.Style=STYLES.Style_ID  
AND  
STYLES.Category='SDG'  
AND  
STYLES.Description LIKE 'white'
```

Find all photographs for project ID 128 with the description containing the word hail

```
SELECT Link_HYP FROM PHOTOS  
WHERE Description LIKE 'hail'  
AND  
Project=128
```

These few simple queries above demonstrate the potential for true information management with the designed database. The number of data queries possible is endless as is the valuable information that may be derived from those queries.

Conclusion

The general contractor business model represented at Suburban Remodeling company is a large, complex mesh of business procedures and processes. Through systematic analysis and implementation, those processes can be moved from the current paper and flat-file based system to a complete relational system managed by a single database.

While limited to the development of the database itself, the scope of this engineering document demonstrates the potential for the development of such a comprehensive data and information system for the benefit of overall business processes.

Given the resources to complete development with a fully functional user interface that maps to current or desired business processes, the General Contractor Database would provide a complete information system beneficial to the tracking of the large array of information that is part of the business practices of Suburban Remodeling.

References

- Alhir, Sinan Si. (1998). *UML in a nutshell*. Sebastopol, CA: O'Reilly.
- Glossary. (Accessed January 17, 2004).
<http://www.keele.ac.uk/depts/cs/Staff/Homes/Mdb3/papers/glossv.htm>
- Pfleeger, Shari Lawrence. (2001). *Software engineering theory and practice*. Singapore: Pearson Education.
- Visualcase. (Accessed January 17, 2004).
http://www.visualcase.com/kbase/data_flow_diagram.htm