

Database Implementation

Group #1

Evan Tyler, Brian Rood, Patrick Niederhauser, Samuel

Emmett

Project overview

- Executive summary: Our business focuses on the production and distribution of Sheet metal. The business is broken down into two separate branches. SMI or Sheet metal industries and Low states. SMI handles all the manufacturing while low states handels the sales and marketing. Although they are separate branches they work hand in hand to provide the best service possible for their consumers. Although they are a well laid out business they lack a proper database. Not having a database is hurting this business, because it lacks the unity across its three locations. Our goal is to create a database that links inventory and client information across all three different locations.

- We all worked equally on the project, So collectively as a group we decided to split the points equally. However we all worked at specific things
 - Patrick Niederhauser - I worked on the final powerpoint and finale report. I also created the original conceptual data model using draw.io with help from my group.
 - Sam Emmett - Provided information on SMI and LowStates, as well as pricing information and insight on operations of the business. Assisted with converting data models to SQL. Also assisted with converting databases from MySQL to Oracle.
 - Evan Tyler - Helped translate data from the SMI and LowStates forums to the actual SQL implementation. Helped with creating and making sure data models were correct and created correctly.
 - Brian Rood - Kept paperwork up to date as well as making corrections, data entry forms and reports, and assisted with data models and SQL conversion.

Contents

1. Cover page -----	1
2. Project Overview-----	2
a. Executive summary-----	2
b. Who did what -----	3
3. Contents-----	4
a. contents-----	5
4. Business description-----	6
a. Name, purpose, summary of activity-----	7
b. Problems, opportunity and objective-----	8
c. Business case, information and data requirements-----	9
d. List of entities-----	10
5. Conceptual data model-----	11
a. Client to order-----	11
b. Order to Job, Order to Employee-----	11
c. Order to warehouse, Employee to warehouse-----	12
d. Warehouse to inventory-Order to inventory-----	13
e. Inventory to item-----	14
f. Item to category-----	15
6. Logical and physical data model-----	16
a. Logical data model-----	16
b. Physical model-----	17
c. Functional dependencies-----	18
d. Functional dependencies-----	19
7. DataBase implementation-----	20
a. Sql codes for creating all normalized tables-----	20

b.	Sql codes for creating all normalized tables-----	21
c.	Sql codes for creating all normalized tables-----	22
d.	Sql codes for creating all normalized tables-----	23
e.	Sql codes for inserting data into each table-----	24
f.	Sql codes for inserting data into each table-----	25
g.	Sql codes for inserting data into each table-----	26
h.	Sql codes for inserting data into each table-----	27
i.	Sql codes for inserting data into each table-----	28
j.	Relationship views-----	29
k.	Model for employee, model for client-----	29
l.	Model for inventory, model for item -----	30
m.	Model for orders, model for jobs-----	31
n.	Model for warehouses-----	32
8.	Designing Form-----	33
a.	Data entry forms -----	33
b.	Data reports -----	34
c.	Data reports-----	35
9.	Appendix-----	36
a.	Team weighting contract-----	36
b.	Team contract-----	37
c.	Team forming contract-----	38
d.	Powerpoint slides-----	39-43

Business Description

- Business Name: Sheet Metal Incorporated & Lowstates
- Purpose of Business
 - SMI(sheet Metal industries) manufactures the sheet metal straight ducts. A contractor will contact Lowstates, a division of SMI industries which handles the design of the duct system and the sale of the duct system to the contractor. Lowstates then will add any necessary fittings,ducts,taps, and collars to the order .Then SMI will go out to the project and install the duct system for the contractor.
- Summary of Business Activities
 - SMI & LowStates have many business activities. The primary business activity of SMI is to produce and manufacture air ducts to be sold in New Jersey, Pittsburg, and delaware. As well as producing air ducts they will also install air ducts at the job site. LowStates deal with the technical aspect of the orders, driving sales and creating marketing opportunities for the business while also dealing with customer service. On top of driving sales and marking Low states also creates CAD drafting and a 3D coordination for SMI to use.

- Problems, Opportunities and objectives
 - The primary issue with LowStates' current method of data storage is syncing inventories across their three locations in New Jersey, Pittsburgh, and Delaware. Their second most pressing issue is organizing and keeping track of local inventories, more specifically order information and whether they have been filled or unfilled, say if Penn Hospital needs 200 ducts and Philly Stadium needs 10,000, and whether or not those orders have been filled.
 - This serves as an opportunity to extensively organize local inventories during the transition to the database and apply any changes they had wanted to improve on their current data storage that they couldn't with their old data storage methods. This could also serve as an opportunity to improve communication between locations in regards to inventory, once the data is properly synced, data and information can be shared much faster and much easier. The same applies to communication between LowStates and SMI, the latter of which would have improved access to LowStates' data.
 - The primary goal of implementing a new database system is to sync warehouse inventory across multiple warehouses. This will make it easier for employees to manage stock allocation to different job sites and orders while simultaneously making it faster, and easier, to access the stock information of on hand inventory.

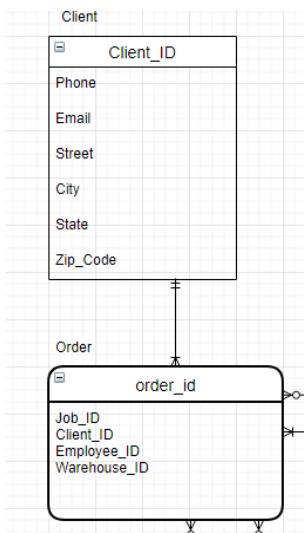
- Business case
 - A new database will allow for employees to access data more easily, while increasing data integrity across multiple locations. This will lead to an overall boost in productivity since warehouses will not need to send emails to request manual inventory checks at different warehouses.
- Information and data requirements
 - In order to achieve the proper entities we would need to collect a multitude of data. Information such as pricing, product, product-type, client information, client location and stock will need to be recorded in order to have accurate entities.
- List of entities(tables)That have been identified
 - CLIENT(Client_ID, Phone, Email, Street, City, State, Zip_Code)
 - ORDER(Order_ID, Job_ID, Client_ID, Employee_ID, Warehouse_ID)
 - EMPLOYEE(Employee_ID, FirstName, LastName, Department, Warehouse_ID)
 - WAREHOUSES(Warehouse_ID, Union_Number, State, Street City, Zip_Code)
 - CATEGORIES (Category_ID,Catgories_Name)
 - ITEM (Item_ID, Category_ID, Item_Price, Product_Name)

- o INVENTORY (Warehouse_ID, Item_ID, Item_Count)
- o JOB(Job_ID, Order_ID, Job_Street, Job_City, Job_State, Zip_Code)

Conceptual Data Model

Client to order

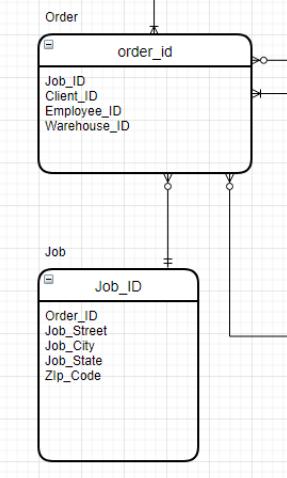
The relationship between Client to Order is one to many, mandatory to mandatory. This is because one and only one client can have many orders. Client is strong because it doesn't depend on another entity for its data. However Order is weak because it depends on client, job, employee and warehouses for its data. Because this is a weak to strong relationship, Order is an ID dependent on Client.



Order to Job

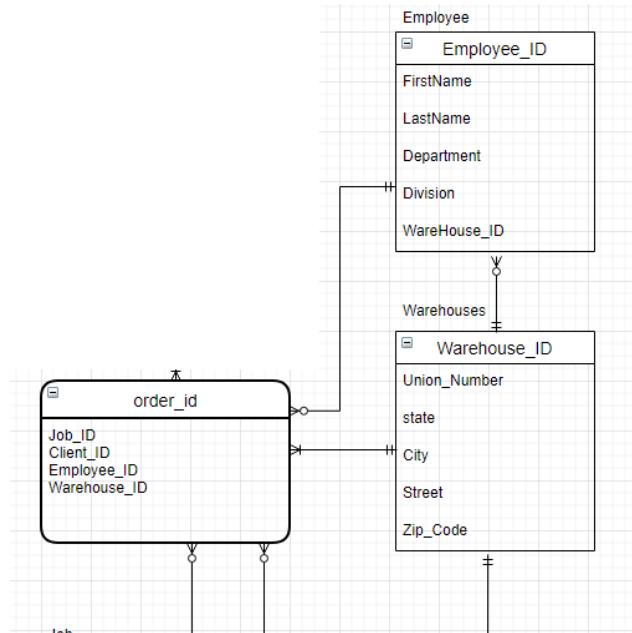
The relationship between Order to Job is many to one, optional to mandatory. This is because you can have many orders, but each order can only come from one and only one job site. Both entities are weak because order relies on employee, warehouse,

client, and job, while job relies on order ID, thus there is no ID dependency. Between them, there is only a non-identifying relationship, as both exist independently of each other.



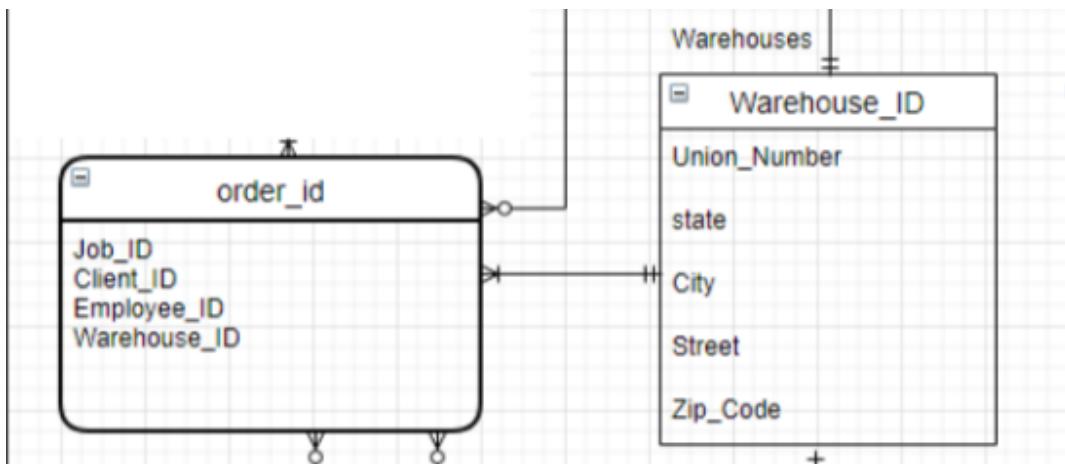
Order to Employee

The relationship between Order to employee is many to one, optional to mandatory so many orders can be made through one employee, likewise the minimal cardinality of the relationship is optional to one. This is because while an order requires that an employee be associated with it, an employee does not need to be associated with an order to be an employee. Order is a weak entity since it relies on employee_ID as a foreign key. Also, Order is ID-dependent on Employee, and Order and Employee have an identifying relationship.



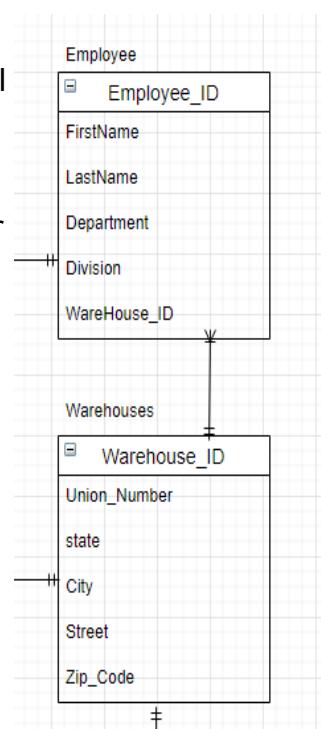
Order to Warehouses

The relationship between Order to Warehouses is many to one, mandatory to mandatory. This is because there will be many orders, but they can only come from one and only one warehouse per order. Order is a weak entity because it relies on employee, warehouse, client, and job, while Warehouses are strong with no reliances. This leads to an ID Dependency of Order on Warehouses. Order and Warehouses have an identifying relationship as Warehouse is necessary for order.



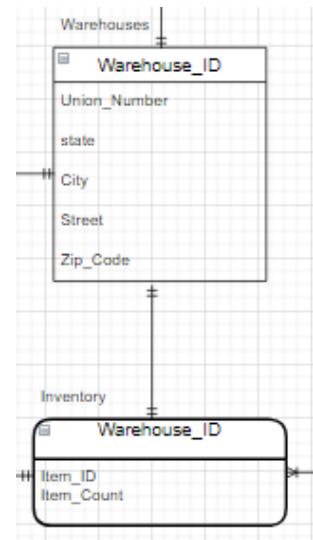
Employee to Warehouse

The relationship between Employee to Warehouse is many to one, optional to mandatory CHANGE TABLE since many Employees can be assigned to one warehouse, but multiple warehouses can not be assigned to a singular employee. Both Employee and Warehouse are strong Entities and, as a result, neither entity is dependent on the other and neither entity has any identifying relationship.



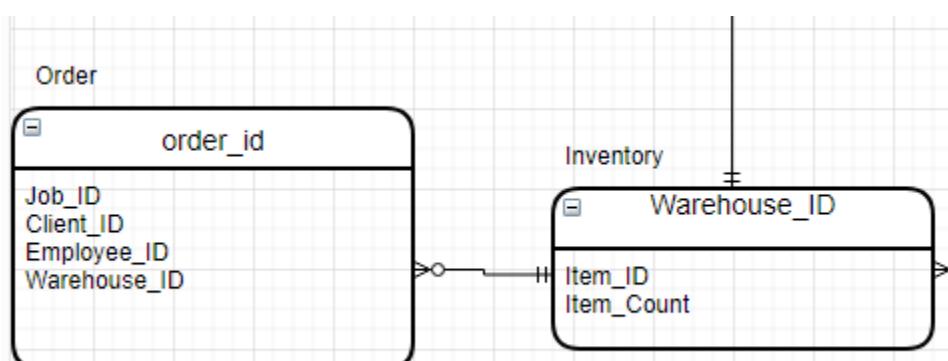
Warehouses to Inventory

The relationship between Warehouses to Inventory is one to one, mandatory to mandatory. This is because each warehouse requires one inventory, and each inventory can only exist in one warehouse. Warehouses is a strong entity with no reliances, and inventory is weak as it relies on Warehouses. This leads to an ID dependency of Inventory on Warehouses, the weak and the strong identifying entity. This also leads to an identifying relationship between the two entities as Inventory is directly defined by Warehouses.



Order to Inventory

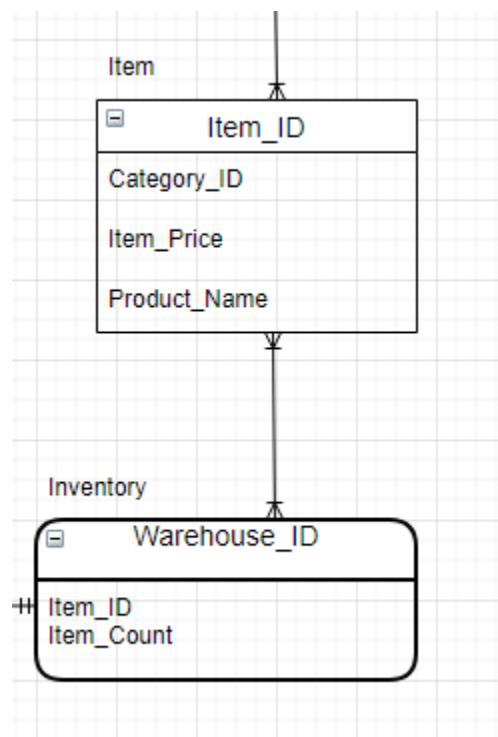
The relationship between Order to Inventory is many to one, optional to mandatory. This is because there could be any number of orders but all of them are from one and only one inventory. Both entities here are weak, since order relies on warehouse, job, client, and employee, and Inventory relies on Warehouses. Since both entities are weak, there is no ID dependency present. The relationship is a non-identifying relationship, as each entity is independent of the other.



Inventory to Item

The relationship between Inventory to Item is many to many, mandatory to optional.

This is because zero to any number of items can belong to any inventory across their respective warehouses. The Inventory entity is weak, as it relies on Warehouses, while Item is strong as an independent entity. Inventory is ID dependent on Item, as it is used in organizing Inventory. Item and Inventory have an identifying relationship, as Inventory is defined by the items contained within it.

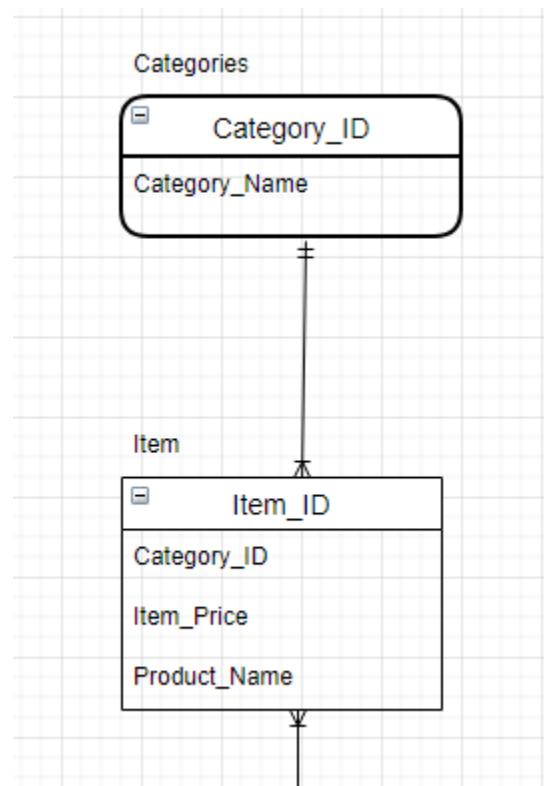


Item to Categories

The relationship between Items to categories is many to one, mandatory to mandatory.

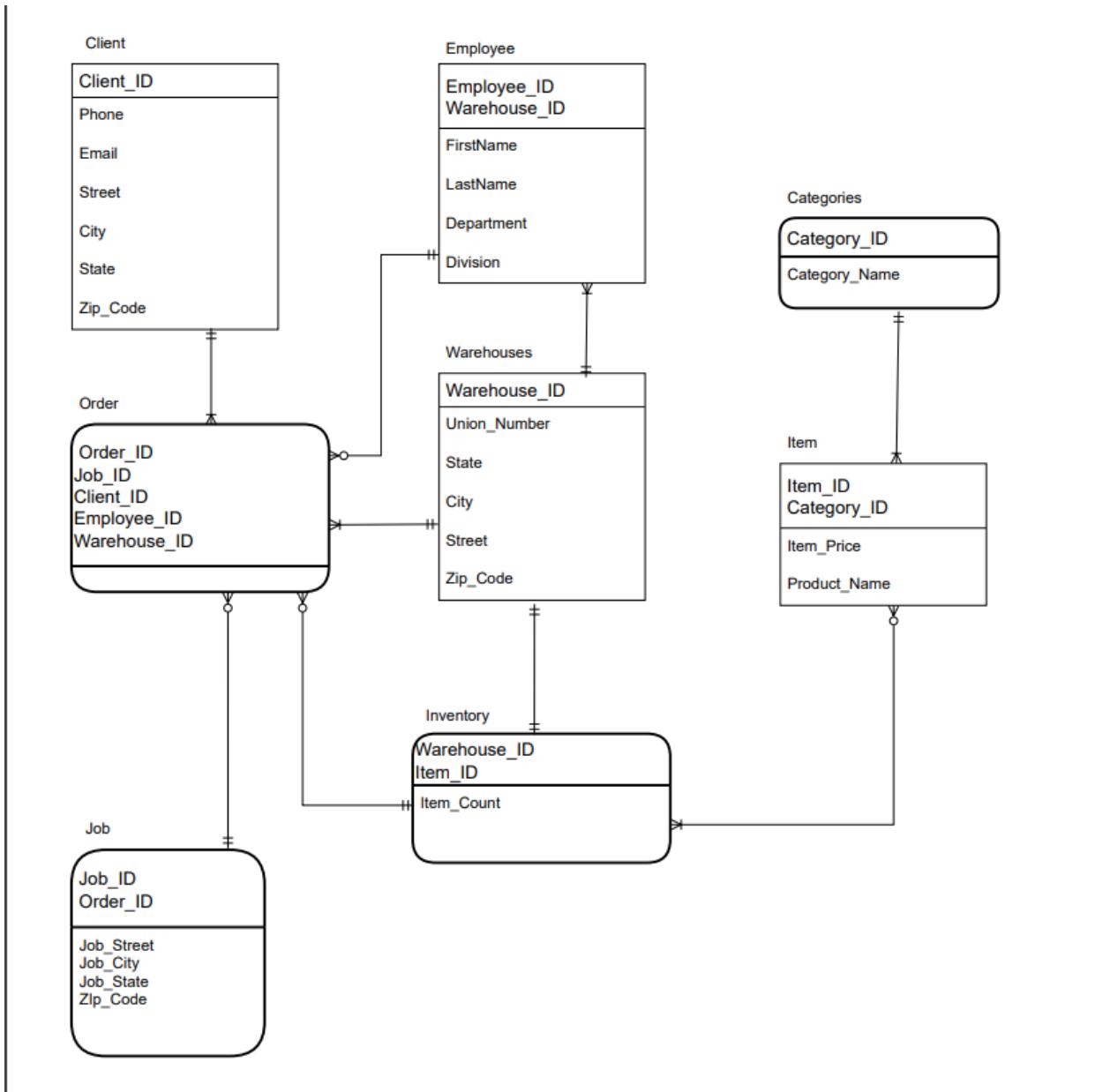
This is because we can have many items but they can only have one category.

Categories is a weak entity because it relies on items for its data. Item is a strong entity because it doesn't rely on any entity for its data. Since categories is a weak entity and Item is a strong entity Categories is an ID dependent on Item. This also infers that Item and Categories have an identifying relationship

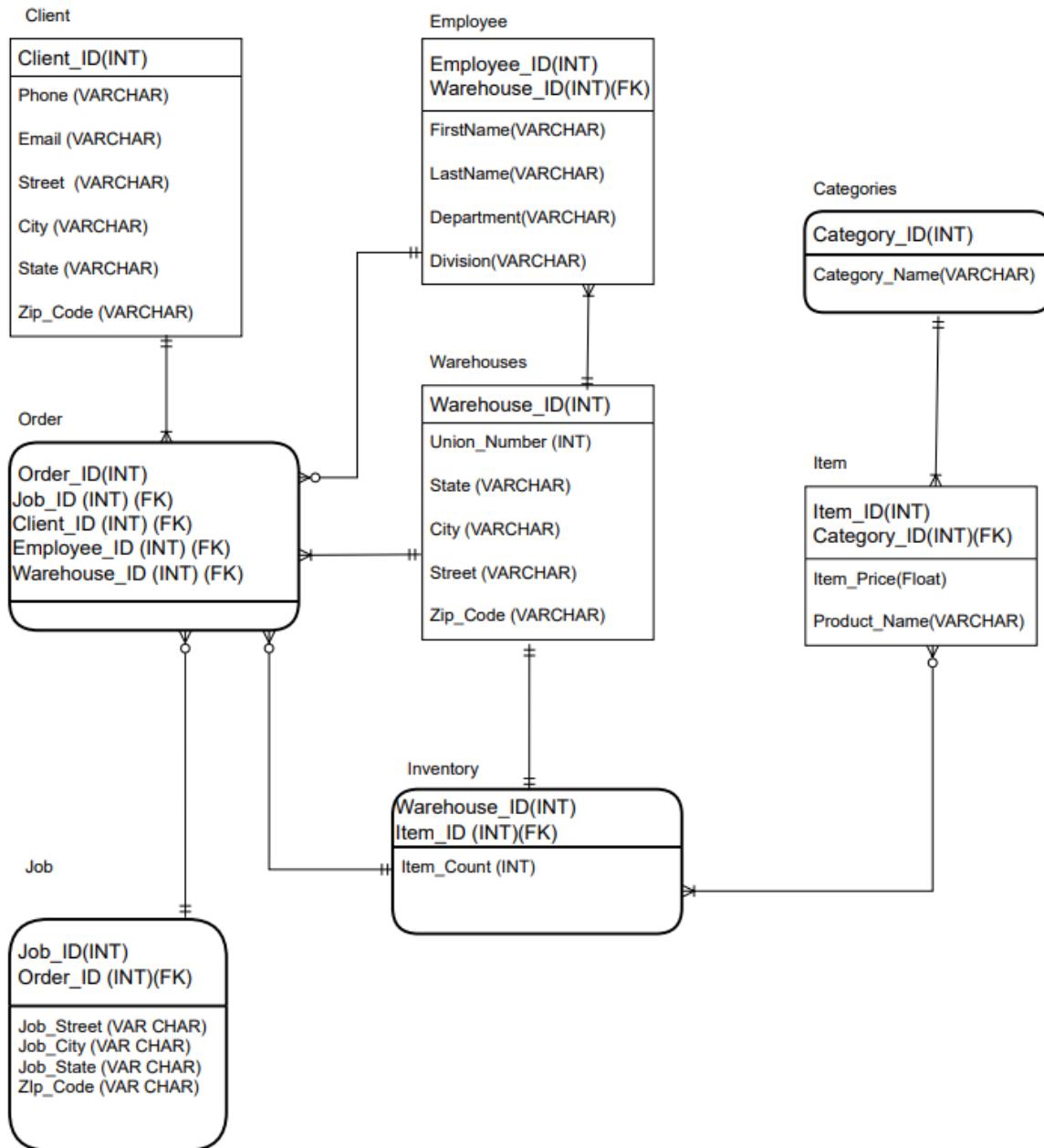


Logical and physical data model

Logical data model



Physical Model



Functional dependencies

Client Table

Client_ID -> (Phone, Email, Street, City, State, Zip_Code)

City -> Zip_Code

State -> City

City -> Street

Employee Table

Employee_ID, Warehouse_ID -> (FirstName, LastName, Department, Division)

Division -> Department

Warehouses Table

Warhouse_ID -> (Union_Number, State, City, Street, Zip_Code)

City -> (Street, Zip_Code)

State -> City

Order Table

Order_ID -> (Job_ID, Client_ID, Employee_ID, Warehouse_ID)

Job_ID -> (Client_ID, Employee_ID, Warehouse_ID)

Employee_ID -> Warehouse_ID

Job Table

Job_ID, Order_ID -> (Job_Street, Job_City, Job_State, Zip_Code)

Job_City -> (Job_Street, Zip_Code)

Job_State -> Job_City

Inventory Table

Warehouse_ID, Item_ID -> Item_Count

Category Table

Category_ID -> Category Name

Item Table

Item_ID, Category_ID -> (Item_Price, Product_Name)

Database Implementation

SQL codes for creating all normalized tables

```
CREATE TABLE Clients(  
    Client_ID INT GENERATED ALWAYS AS IDENTITY  
        (START WITH 1 INCREMENT BY 1),  
    Phone varchar(50) NOT NULL,  
    Email varchar(50) NOT NULL,  
    Street varchar(50)NOT NULL,  
    City varchar(50) NOT NULL,  
    States varchar(50) NOT NULL,  
    Zip_Code varchar(50) NOT NULL,  
    CONSTRAINT Clients_PK PRIMARY KEY (Client_ID)  
);
```

```
CREATE TABLE Categories (  
    Category_ID INT GENERATED ALWAYS AS IDENTITY  
        (START WITH 1 INCREMENT BY 1),  
    Category_Name varchar(200) NOT NULL,  
    CONSTRAINT Cat_PK PRIMARY KEY (Category_ID)  
);
```

```
CREATE TABLE ITEM (  
    Item_ID INT GENERATED AS IDENTITY  
        (START WITH 1 INCREMENT BY 1),  
    Category_ID INT NOT NULL,
```

```

Item_Price Float NOT NULL,
Product_Name varchar(200) NOT NULL,
CONSTRAINT Item_PK PRIMARY KEY (Item_ID),
CONSTRAINT Cat_FK FOREIGN KEY (Category_ID) REFERENCES Categories(Category_ID)
);

```

```

CREATE TABLE Warehouses (
Warehouse_ID INT GENERATED ALWAYS AS IDENTITY
(START WITH 1 INCREMENT BY 1),
Union_Number INT,
States varchar(50) NOT NULL,
City varchar(50) NOT NULL,
Street varchar(50) NOT NULL,
Zip_Code varchar(50) NOT NULL,
CONSTRAINT Warehouse_PK PRIMARY KEY (Warehouse_ID)
);

```

```

CREATE TABLE Inventory (
Warehouse_ID INT GENERATED BY DEFAULT AS IDENTITY
(START WITH 1 INCREMENT BY 1),
Item_ID INT NOT NULL,
Item_Count INT NOT NULL,
CONSTRAINT Inv_PK PRIMARY KEY (Warehouse_Id, Item_ID),
CONSTRAINT Warehouse_FK FOREIGN KEY (Warehouse_ID) REFERENCES
Warehouses(Warehouse_ID),
CONSTRAINT ITEM_FK FOREIGN KEY (Item_ID) REFERENCES Item(Item_ID)
);

```

```
CREATE TABLE Employee(
    Employee_ID INT GENERATED ALWAYS AS IDENTITY
        (START WITH 1 INCREMENT BY 1),
    Warehouse_ID Int NOT NULL,
    FirstName varchar(50) NOT NULL,
    LastName varchar(50) NOT NULL,
    Department varchar(100)NOT NULL,
    Division varchar(100)NOT NULL,
    CONSTRAINT Employee_PK1 PRIMARY KEY (Employee_ID),
    CONSTRAINT Warehouse_FK1 FOREIGN KEY (Warehouse_ID) REFERENCES
Warehouses(Warehouse_ID)
);
```

```
CREATE TABLE Jobs (
    Job_ID INT GENERATED ALWAYS AS IDENTITY
        (START WITH 1 INCREMENT BY 1),
    Job_Street varchar(100) NOT NULL,
    Job_City varchar(100) NOT NULL,
    Job_State varchar(100) NOT NULL,
    Zip_Code varchar(100) NOT NULL,
    CONSTRAINT Jobs_PK PRIMARY KEY (Job_ID)
);
```

```
CREATE TABLE Orders(
    Order_ID INT GENERATED ALWAYS AS IDENTITY
        (START WITH 1 INCREMENT BY 1),
    Job_ID Int NOT NULL,
    Client_ID Int NOT NULL,
```

```
Employee_ID Int NOT NULL,  
Warehouse_ID Int NOT NULL,  
CONSTRAINT Order_PK1 PRIMARY KEY (Order_ID),  
CONSTRAINT Job_FK1 FOREIGN KEY (Job_ID) REFERENCES Jobs(Job_ID),  
CONSTRAINT Client_FK1 FOREIGN KEY (Client_ID) REFERENCES Clients(Client_ID),  
CONSTRAINT Employee_FK1 FOREIGN KEY (Employee_ID) REFERENCES  
Employee(Employee_ID),  
CONSTRAINT Warehouse_FK2 FOREIGN KEY (Warehouse_ID) REFERENCES  
Warehouses(Warehouse_ID)  
);
```

SQL codes for inserting data into each table

-- Data Entry for Categories

```
INSERT into Categories(Category_Name)
```

```
Values ('Duct');
```

```
INSERT into Categories(Category_Name)
```

```
Values ('CPL');
```

```
INSERT into Categories(Category_Name)
```

```
Values ('Connector');
```

```
INSERT into Categories(Category_Name)
```

```
Values ('End');
```

```
INSERT into Categories(Category_Name)
```

```
Values ('Misc');
```

-- Data Entry for Item

```
INSERT into item(Category_ID,Item_Price,Product_Name)
```

```
Values(1,1.58,'Duct');
```

```
INSERT into item(Category_ID,Item_Price,Product_Name)
```

```
Values(2,6.51,'Pipe Cpl');
```

```
INSERT into item(Category_ID,Item_Price,Product_Name)
Values(2,4.72,'Ftg Cpl');
```

```
INSERT into item(Category_ID,Item_Price,Product_Name)
Values(3,15.89,'Short 90');
```

```
INSERT into item(Category_ID,Item_Price,Product_Name)
Values(3,21.70,'Long 90');
```

```
INSERT into item(Category_ID,Item_Price,Product_Name)
Values(3,13.84,'Short 45');
```

```
INSERT into item(Category_ID,Item_Price,Product_Name)
Values(3,18.36,'long 45');
```

```
INSERT into item(Category_ID,Item_Price,Product_Name)
Values(4,8.01,'End cap');
```

```
INSERT into item(Category_ID,Item_Price,Product_Name)
Values(4,34.24,'Damper');
```

```
INSERT into item(Category_ID,Item_Price,Product_Name)
```

```
Values(5,4.94,'start collar');
```

```
INSERT into item(Category_ID,Item_Price,Product_Name)
```

```
Values(5,9.66,'Bell Collar');
```

```
-- Data Entry for Job
```

```
INSERT into jobs(Job_Street,Job_city,Job_State,Zip_Code)
```

```
values ('E Main St','Harlem','NY','10026');
```

```
INSERT into jobs(Job_Street,Job_city,Job_State,Zip_Code)
```

```
values ('John St','Tuckerton','NJ','08087');
```

```
INSERT into jobs(Job_Street,Job_city,Job_State,Zip_Code)
```

```
values ('Elmwood Ln','Marietta','PA','17547');
```

```
INSERT into jobs(Job_Street,Job_city,Job_State,Zip_Code)
```

```
values ('Joey Ln', 'Lancaster','PA', '17601');
```

```
INSERT into jobs(Job_Street,Job_city,Job_State,Zip_Code)
```

```
values ('Fourth Ave','New Castle', 'DE', '19720');
```

```
-- Data Entry for Warehouses
```

```
INSERT INTO Warehouses(Union_Number, States, City, Street, Zip_Code)
```

```
VALUES (11, 'PA', 'Brookeville', '60 progress ST','15825');
```

```
INSERT INTO Warehouses(Union_Number, States, City, Street, Zip_Code)  
VALUES (21, 'DE', 'Clayton', '291 Blackiston RD', '19938');
```

```
INSERT INTO Warehouses(Union_Number, States, City, Street, Zip_Code)  
VALUES (31, 'NJ', 'Tabernacle', '540 Chatsworth RD', '08088');
```

-- Data Entry for Inventory

```
INSERT into inventory(Warehouse_ID,Item_ID,Item_Count)  
values (1,1,754);
```

```
INSERT into inventory(Warehouse_ID,Item_ID,Item_Count)  
values (1,2,11);
```

```
INSERT into inventory(Warehouse_ID,Item_ID,Item_Count)  
values (1,3,123);
```

```
INSERT into inventory(Warehouse_ID,Item_ID,Item_Count)  
values (1,4,234);
```

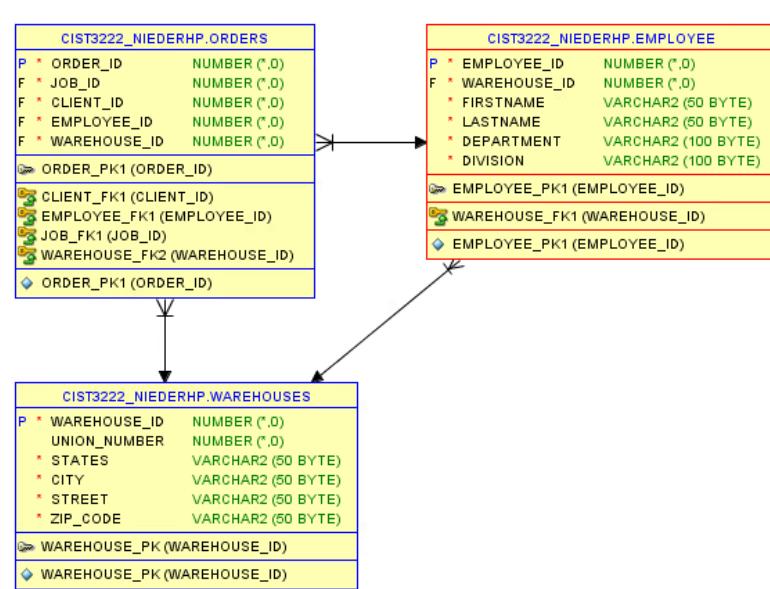
```
INSERT into inventory(Warehouse_ID,Item_ID,Item_Count)  
values (1,5,245);
```

```
INSERT into inventory(Warehouse_ID,Item_ID,Item_Count)
values (1,6,13);
```

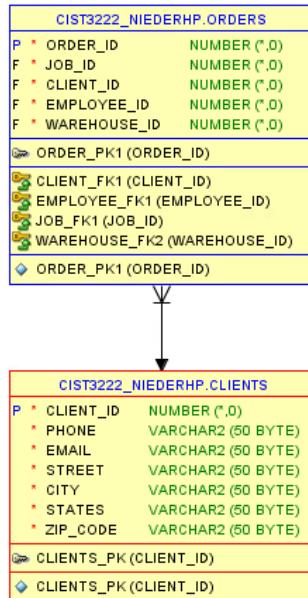
```
INSERT into inventory(Warehouse_ID,Item_ID,Item_Count)
values (1,7,323);
```

Relationships view

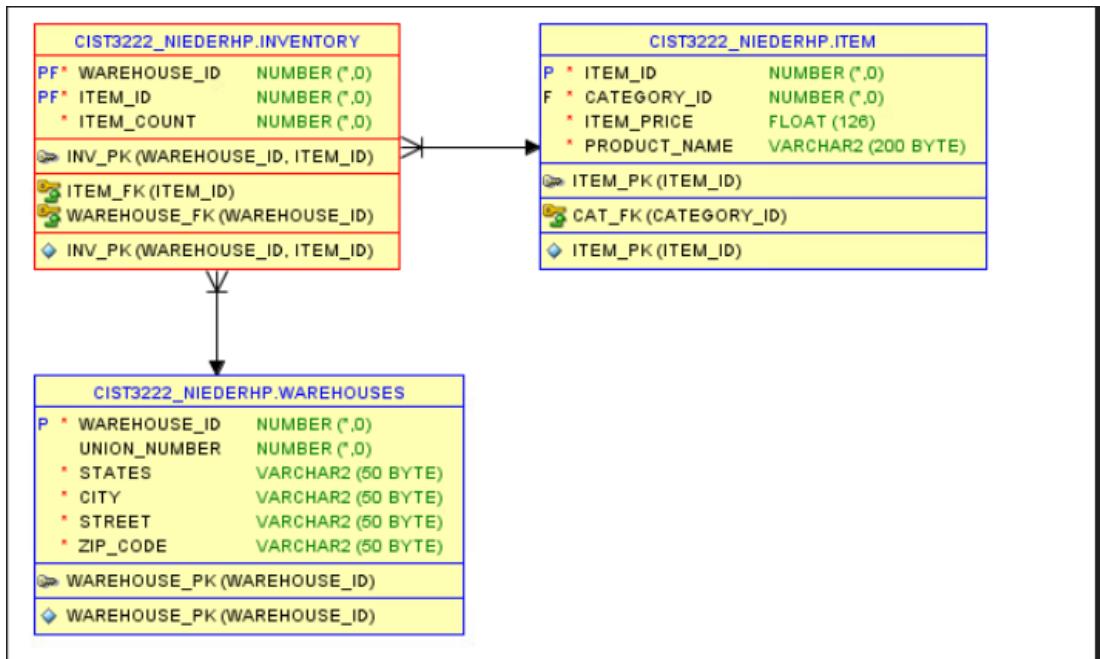
Model for employee



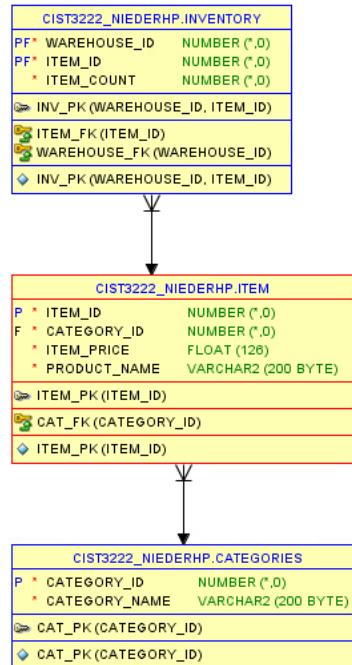
Model for client



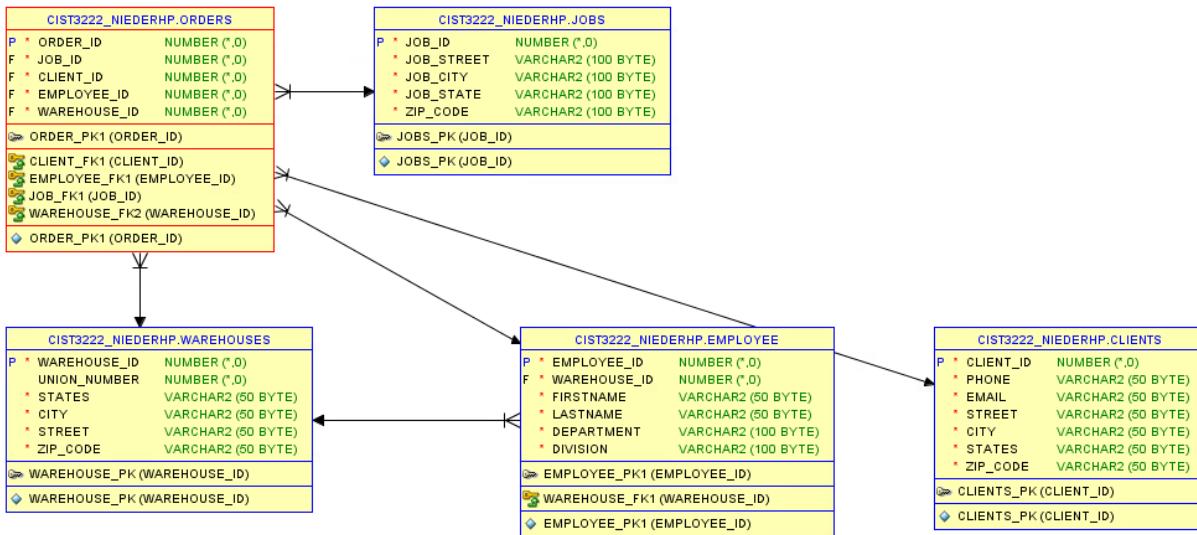
Model for inventory



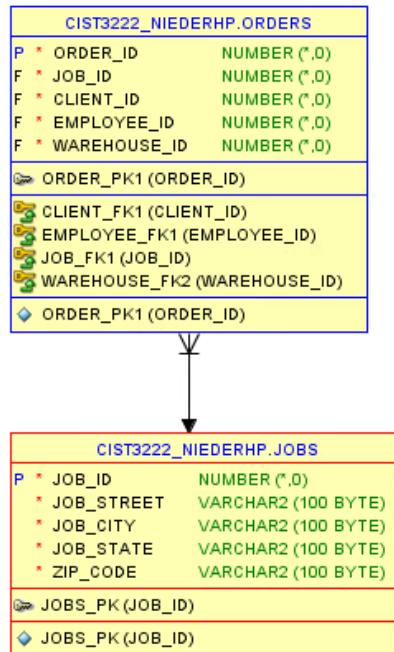
Model for item



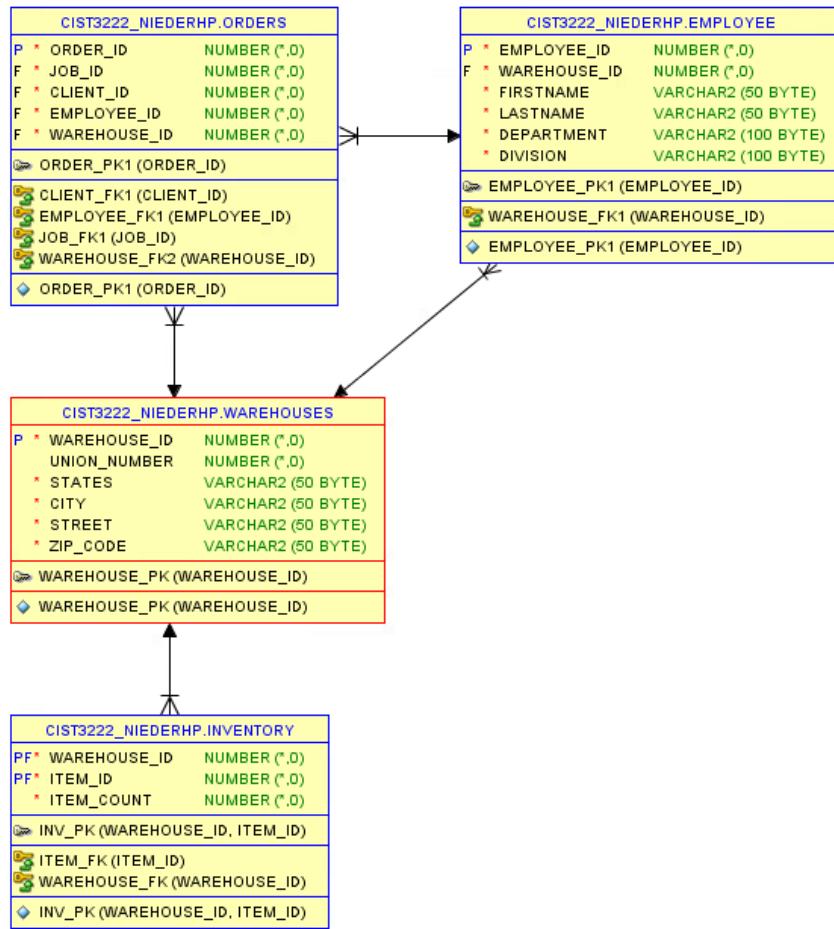
Model for orders



Model for jobs



Model for warehouse



Designing Form

Data entry forms

This form takes input from the user and implements it into the Employee table

Employee Form

Employee ID:	72824139
Warehouse ID:	11
First Name:	Jerry
Last Name:	John
Department:	Sales
Division:	Low States

1 of 6

New
Delete
Restore
Find Prev
Find Next
Criteria
Close

This form takes input from the user and implements it into the Client table

Client Form

Phone:	509-298-1991
Email:	Ckasano@Gmail.com
Street:	35 Claridge dr
City:	Jackson
State:	NJ
Zip Code:	18527

1 of 7

New
Delete
Restore
Find Prev
Find Next
Criteria
Close

This form takes input from the user and implements it into the order table

Order Form

Job ID:	101
Client ID:	1
Employee ID:	72824139
Warehouse ID:	11

?

X

1 of 6

New

Delete

Restore

Find Prev

Find Next

Criteria

Close

Data reports

Data report for the employee table

Employee Form
SMI and Lows States Industries

Employee ID	Warehouse ID	First Name	Last Name	Department	Division
72824139	11	Jerry	John	Sales	Low States
50925192	11	Betty	Slocker	Accounting	Low States
73782312	21	Helen	Kenin	Sales	Low States
85177823	21	Allen	Jeremy	Sheet Metal Worker	SMI
75454213	31	Melissa	Sekura	Collections	SMI
86231239	21	Violetta	Genssy	Accounting	Low States

Data report for the Client Table

Client information Form Report SMI and Lows States Industries						
Phone	Email	Street	City	State	Zip Code	
509-298-1991	Ckasano@Gmail.com	35 Claridge dr	Jackson	NJ	18527	
732-240-5992	Yisirpina518@Gmail.com	126 Ford Ave	Neptune	NJ	17720	
630-215-5821	GossipGirlFan@Gmail.com	1001 California Ave	Pittsburgh	PA	15290	
412-422-3790	sc10010030n@Gmail.com	2008 Murray Ave	Pittsburgh	PA	15217	
302-434-3200	9vbsby20110@gmail.com	2230 Hessler Blvd	New Castle	DE	19720	
800-275-8777	xmishae@gmail.com	347 S Bouquet St.	Pittsburgh	PA	15213	
302-655-8077	7abdou@gmail.com	700 W 4th St,	Wilmington	DE	19801	

Data report for the Order Table

Order Form SMI and Lows States Industries				
Job ID	Client ID	Employee ID	WareHouse ID	
101	1	72824139	11	
101	2	50925192	11	
201	3	73782312	21	
201	4	85177823	21	
301	5	75454213	31	
201	6	86231239	21	

Appendix

CIST 3222 TEAM WEIGHTING CONTRACT
Team Project # 1

By signing this form you agree to the following:

This Team Weighting Contract creates a binding contract between the individual members of the Team and the faculty of CIST 3222. As the undersigned of sound mind and body, I agree that this is a fair and accurate representation to the amount of work contributed by my Team members and myself. I agree to bear the consequences of this binding contract. In acceptance of this signed contract, the faculty of CIST 3222 promises to accept this form as is and refuses any grievances in conjunction therein. Upon my signature and acceptance of this form, I waive my right to enter into further negotiations regarding this form or alter the contents of this form.

Any form of fraud will render this contract null and void. Any misrepresentation in amount of work contributed to the Team will constitute fraud and void the contract. If the contract is void, then credit distribution will be at the discretion of your Instructor. Any instance of fraud forfeits your rights as a Team to determine the rationing of credits. The Team Weighting Contract Directions and Team Forming Contract shall be interpreted as a binding part of this contract.

DO NOT SIGN A BLANK FORM.

CIST 3222 Database Systems
Date: 11-30-21

Team member 1 name (type)	Credits awarded (type)	I, <u>P.N</u> , (initials signature) have read the Team Weighting Contract Directions and this Team Weighting Contract in full.
<u>Patrick Niederhauser</u>	<u>100</u>	
Team member 1 (signature)		
<u>Pn</u>		
Team member 2 name (type)	Credits awarded (type)	I, <u>S.E</u> , (initials signature) have read the Team Weighting Contract Directions and this Team Weighting Contract in full.
<u>Samuel Emmett</u>	<u>100</u>	
Team member 2 (signature)		
<u>Samuel Emmett</u>		
Team member 3 name (type)	Credits awarded (type)	I, <u>E.T</u> , (initials signature) have read the Team Weighting Contract Directions and this Team Weighting Contract in full.
<u>Evan Tyler</u>	<u>100</u>	
Team member 3 (signature)		
<u>Evan Tyler</u>		
Team member 4 name (type)	Credits awarded (type)	I, <u>B.R</u> , (initials signature) have read the Team Weighting Contract Directions and this Team Weighting Contract in full.
<u>Brian Rood</u>	<u>100</u>	
Team member 4 (signature)		
<u>B Rood</u>		

Using the following scale, mark the response that best represents your Team's agreement with the statement below:

Overall, our Team worked well together.

Strongly Disagree Disagree Undecided Agree Strongly Agree

We, the Team, do hereby agree to all of the pages in this document (pages 1-10).
We, the Team, agree to be bound by these rules through the entire semester. We,
the Team, also further agree that we have completely read and understood this
document hereto signed today.

Course: CIST 3222 Database Systems

Date: 09-30-2021

Instructor: Quynh Nguyen

Team member 1 name (print)	I, <u>BN</u> , (initials signature) have read the Team Forming Contract in full, pages 1- 10.
Team member 1 (signature)	<u>BN</u>
Team member 2 name (print)	I, <u>SE</u> , (initials signature) have read the Team Forming Contract in full, pages 1- 10.
Team member 2 (signature)	<u>Samuel Emmett</u>
Team member 3 name (print)	I, <u>BR</u> , (initials signature) have read the Team Forming Contract in full, pages 1- 10.
Team member 3 (signature)	<u>Brian Rood</u>
Team member 4 name (print)	I, <u>ET</u> , (initials signature) have read the Team Forming Contract in full, pages 1- 10.
Team member 4 (signature)	<u>Evan Tyler</u>

Team Forming Contract

- (1) The CIST 3222 student team number #1 whose members are Patrick Niederhauser, Samuel Emmett, Brain Rood, and Evan Tyler ("Team") do hereby enter into this agreement on September 30th, 2021.
- (2) This agreement will govern all of the activities of said Team during this semester.
- (3) This agreement is merged with the pages initialed and all documents referenced by this agreement.
- (4) This contract is bound to the jurisdiction of <<Stockton University>>.
- (5) Team's Instructor will be the arbiter and mediator for any dispute arising in Team throughout this semester.
- (6) Team is joint and severally liable for the completion of this Team Forming Contract and all other necessary and required documents throughout the semester.
- (7) One act may violate more than one rule. In this case, the consequences for all rules violated may be instituted. Documented emergencies will excuse the violation of rules reasonably related to that emergency.

PowerPoint

CIST 3222 Section 1- Database System Database Implementation Finale presentation (SMI & Lowstates)

Group 1
Evan Tyler, Brian Rood, Patrick Niederhauser, Samuel Emmett

Business Overview

SMI & Lowstates

- Manufacturing
- Sales/Design
- Installation



Summary of Business Activities

Specifics business activities of SMI

- To produce and manufacturer air ducts
- To install air ducts

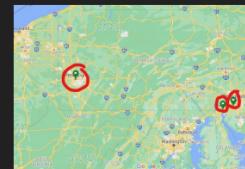


Specific business activities of LowStates

- Sales and marketing
- Customer service
- To produce cad and 3d models

Business Problems

- Syncing inventories across LowStates' three locations
- Organizing local inventories



Opportunities

- Extensively organize local inventories during the transition
- Improve communication between LowStates locations
- Improve communication between SMI and LowStates



Objectives

- Sync inventories
- Manage jobsite allocation
- Manage orders



Business Case

- No current system
- New system would:
 - Make data more accessible
 - Increases data integrity across locations
 - Increase overall productivity



Information and Data Requirement

Information Needed

- Pricing
- Product
- Product-type
- Location
- Client information

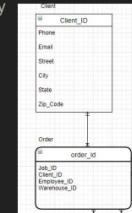


Entities

- CLIENT(Client_ID, Phone, Email, Street, City, State, Zip_Code)
- ORDER(Order_ID, Job_ID, Client_ID, Employee_ID, Warehouse_ID)
- EMPLOYEE(Employee_ID, FirstName, LastName, Department, Warehouse_ID)
- WAREHOUSES(Warehouse_ID, Union_Number, State, Street, City, Zip_Code)
- CATEGORIES(Category_ID, Category_Name)
- ITEM(Item_ID, Category_ID, Item_Price, Product_Name)
- INVENTORY(Warehouse_ID, Item_ID, Item_Count)

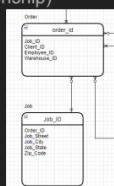
Conceptual Data Model Client to order

- Relationship: One to many, mandatory to mandatory
- Client: Strong, Order: Weak
- Order is ID dependent on client
- Order & Client have an identifying relationship



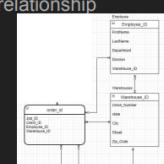
Order to Job

- Relationship: Many to one, optional to mandatory
- Order: Weak, Job: Weak
- They are not ID dependent (weak to weak relationship)
- They have a non-identifying relationship



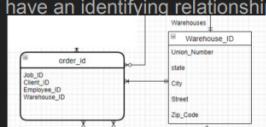
Order to employee

- Relationship: Many to one, optional to mandatory
- Order: weak, Employee: strong
- order is ID dependent on employee
- Order & Employee have an identifying relationship



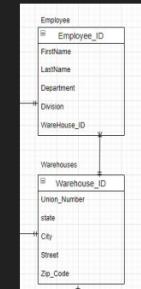
Order to Warehouses

- Relationship: Many to one, mandatory to mandatory
- Order: weak, Warehouses:Strong
- Order is ID dependent on Warehouse_ID
- Order & Warehouses have an identifying relationship



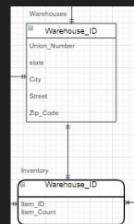
Employee to Warehouses

- Relationship: Many to one, mandatory to mandatory
- Employee:strong, Warehouses:strong
- Warehouse_ID and Employee_ID are not ID dependent on each other
- Employee & Warehouses don't have an identifying relationship



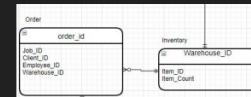
Warehouses to inventory

- Relationship: one to one, mandatory to mandatory
- Warehouses:Strong, Inventory:weak
- Inventory is an ID dependent on warehouses
- Warehouses & Inventory have an identifying relationship



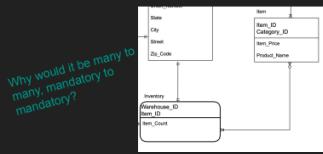
Order to Inventory

- Relationship: Many to one, optional to mandatory
- Order: weak, Inventory: weak
- order is not ID dependent on Inventory
- Order & Inventory don't have an identifying relationship

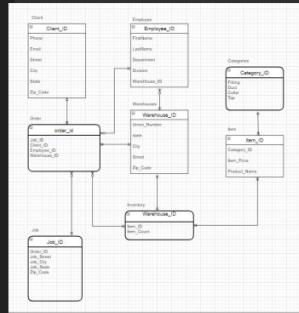


Inventory to Item

- Relationship: Many to many, mandatory to optional
- Inventory: weak, Item: strong
- Inventory is ID dependent on Item
- Item & Inventory have an identifying relationship

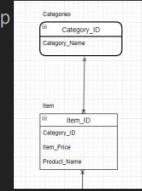


Conceptual data model

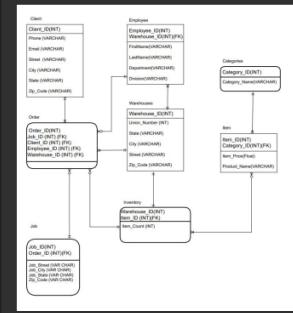


Item to categories

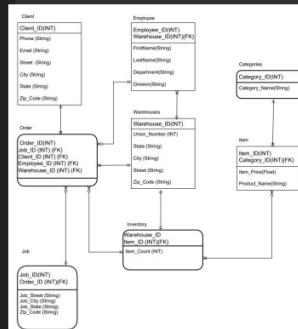
- Relationship: Many to one, mandatory to mandatory
- Item: strong, Categories: weak
- Categories is ID dependent on Item
- Item & Inventory have an identifying relationship



Logical data model

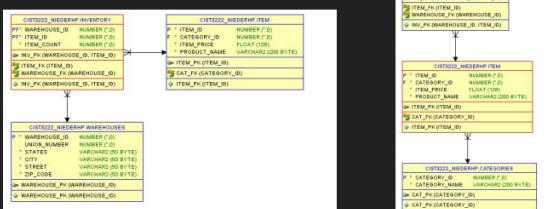


Physical data model



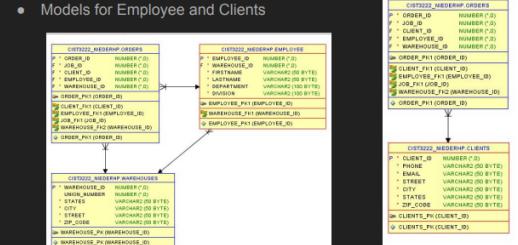
Database implementation

- Models for Inventory and item



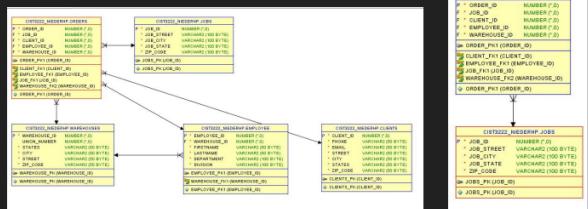
Database implementation

- Models for Employee and Clients



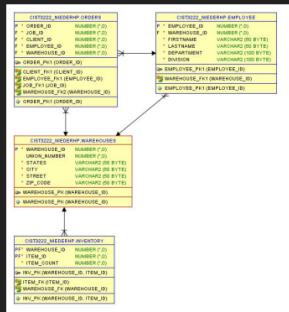
Database implementation

- Models for orders and Jobs



Database implementation

- Model for Warehouse



More tables

This is the table for Inventory

	Warehouse_ID	Item_ID	Item_Count
1	1	754	
1	2	11	
1	3	123	
1	4	254	
1	5	245	
1	6	13	
1	7	323	
1	8	233	
1	9	52	
1	10	23	
1	11	15	
1	12	71	
1	13	22	
1	14	77	
2	3	78	
2	4	29	
2	5	44	
2	6	75	
2	7	54	
2	8	9	
2	9	19	
2	10	25	
2	11	15	
2	12	11	
2	13	22	
2	14	26	
2	15	3	
2	16	25	
2	17	4	
2	18	6	
2	19	5	
2	20	23	
2	21	15	
2	22	232	
3	1	232	
3	2	26	
3	3	3	
3	4	25	
3	5	23	
3	6	656	
3	7	23	
3	8	52	
3	9	53	
3	10	29	
3	11	130	
3	12	656	
3	13	11	
3	14	22	
3	15	26	
3	16	25	
3	17	6	
3	18	5	
3	19	23	
3	20	656	

Database implementation + tables

- This is the table for Client

Client_ID	Phone	Email	Street	City	State	Zip_Code
1	509-298-1991	Claesano@gmail.com	35 Clarendon dr	Jackson	NJ	18527
2	732-240-9992	Yarpeas18@gmail.com	125 Ford Ave	Neptune	NJ	17720
3	430-222-1234	Sc10010030@gmail.com	123 Main St	Hanover	PA	15217
4	412-422-3790	Sc10010030@gmail.com	2008 Murray Ave	Pittsburgh	PA	15217
5	302-434-3200	91buby20110@gmail.com	2230 Hessler Blvd	New Castle	DE	19720
6	800-279-9777	79elbow@gmail.com	3475 Bouquet St	Pittsburgh	PA	15213
7	302-655-0077	79elbow@gmail.com	120 W 4th St	Pittsburgh	PA	15201

- This is the table for Employee

Employee_ID	Warehouse_ID	FirstName	LastName	Department	Division
1	1	Jerry	John	Sales	Low States
2	1	Betty	Slecker	Sales	Low States
3	2	Allen	Jeremy	Sheet Metal Worker	SME
4	2	Melissa	Sekura	Collections	SME
5	3	Helen	Karin	Sales	Low States
6	3	Violetta	Gennsy	Accounting	Low States

More tables

- This is the table for Job

Job_ID	Job_Street	Job_City	Job_State	Zip_Code
1	E Main St	Harlem	NY	10026
2	John St	Tuderton	NJ	08087
3	Elmwood Ln	Marieetta	PA	17547
4	Joey Ln	Lancaster	PA	17601
5	Fourth Ave	New Castle	DE	19720

- This is the table for Orders

Order_ID	Job_ID	Client_ID	Employee_ID	Warehouse_ID
1	1	2	1	1
2	2	2	1	2
3	3	3	2	3
4	4	3	1	3
5	5	5	2	2

Data Entry Forms

- This form takes input from the user and implements it into the Client table

Data Entry Forms

- This form takes input from the user and implements it into the order table

<h3>Data Reports</h3> <ul style="list-style-type: none"> Employee report Example <p>Employee Form SMI and Lows States Industries</p> <table border="1"> <thead> <tr> <th>Employee ID</th> <th>Warehouse ID</th> <th>First Name</th> <th>Last Name</th> <th>Department</th> <th>Division</th> </tr> </thead> <tbody> <tr><td>72824139</td><td>11</td><td>Jerry</td><td>John</td><td>Sales</td><td>Low States</td></tr> <tr><td>50925192</td><td>11</td><td>Betty</td><td>Slocker</td><td>Accounting</td><td>Low States</td></tr> <tr><td>73782312</td><td>21</td><td>Helen</td><td>Kenin</td><td>Sales</td><td>Low States</td></tr> <tr><td>85177823</td><td>21</td><td>Allen</td><td>Jeremy</td><td>Sheet Metal Worker</td><td>SMI</td></tr> <tr><td>75454213</td><td>31</td><td>Melissa</td><td>Sekura</td><td>Collections</td><td>SMI</td></tr> <tr><td>86231239</td><td>21</td><td>Violetta</td><td>Genssy</td><td>Accounting</td><td>Low States</td></tr> </tbody> </table>	Employee ID	Warehouse ID	First Name	Last Name	Department	Division	72824139	11	Jerry	John	Sales	Low States	50925192	11	Betty	Slocker	Accounting	Low States	73782312	21	Helen	Kenin	Sales	Low States	85177823	21	Allen	Jeremy	Sheet Metal Worker	SMI	75454213	31	Melissa	Sekura	Collections	SMI	86231239	21	Violetta	Genssy	Accounting	Low States	<h3>Data Reports</h3> <ul style="list-style-type: none"> Client report Example <p>Client Information Form Report SMI and Lows States Industries</p> <table border="1"> <thead> <tr> <th>Phone</th> <th>Email</th> <th>Street</th> <th>City</th> <th>State</th> <th>Zip Code</th> </tr> </thead> <tbody> <tr><td>509-298-1991</td><td>OkasanogGmail.com</td><td>35 Claridge dr</td><td>Jackson</td><td>NJ</td><td>18527</td></tr> <tr><td>732-240-5992</td><td>Yslipina518@Gmail.com</td><td>126 Ford Ave</td><td>Neptune</td><td>NJ</td><td>17720</td></tr> <tr><td>630-215-5821</td><td>GossipGifAn@Gmail.com</td><td>1001 California Ave</td><td>Pittsburgh</td><td>PA</td><td>15290</td></tr> <tr><td>412-422-3790</td><td>sc10010030n@Gmail.com</td><td>2088 Murray Ave</td><td>Pittsburgh</td><td>PA</td><td>15217</td></tr> <tr><td>302-434-3200</td><td>9vbsby20110@gmail.com</td><td>2230 Hessler Blvd</td><td>New Castle</td><td>DE</td><td>19720</td></tr> <tr><td>800-275-8777</td><td>xmishae@gmail.com</td><td>347 S Bouquet St.</td><td>Pittsburgh</td><td>PA</td><td>15213</td></tr> <tr><td>302-655-8077</td><td>7abdou@gmail.com</td><td>700 W 4th St.</td><td>Wilmington</td><td>DE</td><td>19801</td></tr> </tbody> </table>	Phone	Email	Street	City	State	Zip Code	509-298-1991	OkasanogGmail.com	35 Claridge dr	Jackson	NJ	18527	732-240-5992	Yslipina518@Gmail.com	126 Ford Ave	Neptune	NJ	17720	630-215-5821	GossipGifAn@Gmail.com	1001 California Ave	Pittsburgh	PA	15290	412-422-3790	sc10010030n@Gmail.com	2088 Murray Ave	Pittsburgh	PA	15217	302-434-3200	9vbsby20110@gmail.com	2230 Hessler Blvd	New Castle	DE	19720	800-275-8777	xmishae@gmail.com	347 S Bouquet St.	Pittsburgh	PA	15213	302-655-8077	7abdou@gmail.com	700 W 4th St.	Wilmington	DE	19801
Employee ID	Warehouse ID	First Name	Last Name	Department	Division																																																																																						
72824139	11	Jerry	John	Sales	Low States																																																																																						
50925192	11	Betty	Slocker	Accounting	Low States																																																																																						
73782312	21	Helen	Kenin	Sales	Low States																																																																																						
85177823	21	Allen	Jeremy	Sheet Metal Worker	SMI																																																																																						
75454213	31	Melissa	Sekura	Collections	SMI																																																																																						
86231239	21	Violetta	Genssy	Accounting	Low States																																																																																						
Phone	Email	Street	City	State	Zip Code																																																																																						
509-298-1991	OkasanogGmail.com	35 Claridge dr	Jackson	NJ	18527																																																																																						
732-240-5992	Yslipina518@Gmail.com	126 Ford Ave	Neptune	NJ	17720																																																																																						
630-215-5821	GossipGifAn@Gmail.com	1001 California Ave	Pittsburgh	PA	15290																																																																																						
412-422-3790	sc10010030n@Gmail.com	2088 Murray Ave	Pittsburgh	PA	15217																																																																																						
302-434-3200	9vbsby20110@gmail.com	2230 Hessler Blvd	New Castle	DE	19720																																																																																						
800-275-8777	xmishae@gmail.com	347 S Bouquet St.	Pittsburgh	PA	15213																																																																																						
302-655-8077	7abdou@gmail.com	700 W 4th St.	Wilmington	DE	19801																																																																																						
<h3>Data Reports</h3> <ul style="list-style-type: none"> Order report Example <p>Order Form SMI and Lows States Industries</p> <table border="1"> <thead> <tr> <th>Job ID</th> <th>Client ID</th> <th>Employee ID</th> <th>WareHouse ID</th> </tr> </thead> <tbody> <tr><td>101</td><td>1</td><td>72824139</td><td>11</td></tr> <tr><td>101</td><td>2</td><td>50925192</td><td>11</td></tr> <tr><td>201</td><td>3</td><td>73782312</td><td>21</td></tr> <tr><td>201</td><td>4</td><td>85177823</td><td>21</td></tr> <tr><td>301</td><td>5</td><td>75454213</td><td>31</td></tr> <tr><td>201</td><td>6</td><td>86231239</td><td>21</td></tr> </tbody> </table>	Job ID	Client ID	Employee ID	WareHouse ID	101	1	72824139	11	101	2	50925192	11	201	3	73782312	21	201	4	85177823	21	301	5	75454213	31	201	6	86231239	21	<h3>Any Questions?</h3> 																																																														
Job ID	Client ID	Employee ID	WareHouse ID																																																																																								
101	1	72824139	11																																																																																								
101	2	50925192	11																																																																																								
201	3	73782312	21																																																																																								
201	4	85177823	21																																																																																								
301	5	75454213	31																																																																																								
201	6	86231239	21																																																																																								