**Table of Contents**

Page #

1. Requirement/Overview
2. ERD Model
3. Relational Schema
4. Data Dictionary
   1. References (limit 1 page)
5. DDL
   1. Budget Justification
6. Queries
7. Stored Procedures
8. Application Code
9. Appendix

**Requirement Overview**

Description: A database system about how postal management system works. The management system focuses on manual processes for mail and package intake and delivery, usually with a front desk worker or back office employee receiving.

Below are the major entities that will interact to make this system working.

Customer: For every person who comes in as a Sender, Postal Management will capture the Customer’s full name, phone numbers, emails, address details including address line, city, state, country and zip code and auto generated customer Id to unique identify any customer. We also capture the Receiver details as a different customer with the same attributes as defined for Sender.

Transaction: For every item that a Customer/Sender requests to send will be a enclosed in terms of a transaction that is unique for a particular Sender, Receiver and item details to be send across.

For each item to be delivered, we will record the itemType, itemCategory, deliveryType, transaction start date, sender’s detail, receiver’s detail and unique id of the employee who performed this transaction and the store id where this transaction is performed.

ItemType can take different values like Delicate, Document, Standard.

ItemCategory can take different values like Small, Medium, Large.

Delivery Type can take different values like Standard, Overnight, and Urgent.

Depending upon the details entered system will calculate the charges and the delivery date.

Store: We will record every store name, and address details including the address line, city, state, zip code and country and an auto generated unique id that will uniquely identify the store. A store can employ multiple employees.

Employee: We will capture all the employees working in the different store throughout the country. There may be multiple employees working for a store. Every employee will have SSN that will uniquely identify each employee. Also, we will capture employee full name, salary, emails, phone numbers and address details including address line, city, state, zip code, country.

An employee can either be a Full time employee or part time employee with fixed annual salary or it can be a part time with hourly salary.

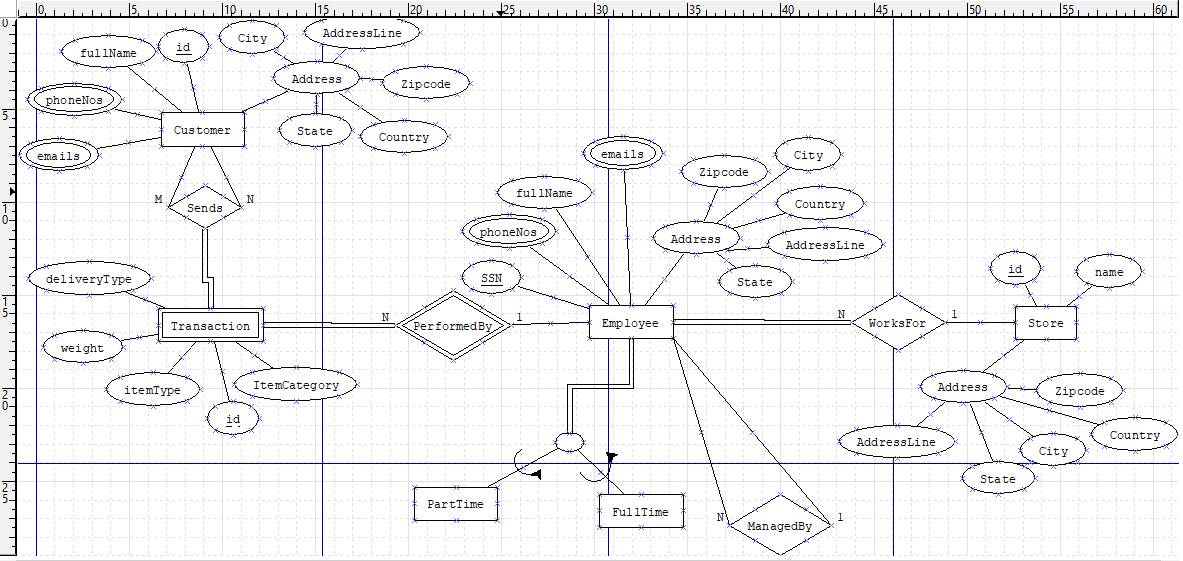
Manager: There is a hierarchy among the employees. An employee can report to a Manager, and Manager can have multiple employees working under him.

Services provided:

1. Create, update, view and delete Store details.
2. Add, update, view and delete employee(s) to Store(s).
3. Manage employee and manager hierarchy. Assign, un-assign manager and employee.
4. Sender can come in make a transaction in the store that will involve shipping an item from one store to the receiver’s address.
5. Store manager manages multiple employees at a store and oversees the transaction.
6. Sender can update the receiver’s address for the item shipment.
7. Sender can request the delivery type (Standard or Priority) only before the item is not assigned a stamp. Sender can view the status of transaction.
8. Calculate the bill of transaction for the sender based on his/her preferences like Item type (ItemType, ItemCategory, DeliveryType)
9. Generate different report based on the different criteria:

* Number of transactions performed in a particular store between a date range.
* Number of transactions performed by a particular employee in a specific store.
* Number of items shipped from a particular Customer.

**ERD Model**



**Relational Schema**

Customer(id, fullName, AddressAddressLine, AddressCity, AddressState, AddressCountry, AddressZipcode)

CustomerAlias1(id, phoneNos)

CustomerAlias2(id, emails)

Transaction(CustomerId, id, SSN, itemType, itemcategory, weight, deliveryType)

Sends(CustomerId, TransactionId)

Employee(SSN, StoreId, ManagedBy, fullName, AddressAddressLine, AddressCity, AddressState, AddressCountry, AddressZipcode)

EmployeeAlias1(SSN, phoneNos)

EmployeeAlias1(SSN, emails)

PartTime(EmployeeSSN)

FullTime(EmployeeSSN)

Store(id, name, AddressAddressLine, AddressCity, AddressState, AddressCountry, AddressZipcode)

**DDL SQL**

**customer**

create table customer (id number(10,0) not null, addreess\_line varchar2(255 char), city varchar2(255 char), country varchar2(255 char), name varchar2(255 char) not null, state varchar2(255 char), zip\_code varchar2(255 char), primary key (id))

**customer\_emails**

create table customer\_emails (customer\_id number(10,0) not null, emailid varchar2(255 char))

**customer\_phone\_numbers**

create table customer\_phone\_numbers (customer\_id number(10,0) not null, phone\_number varchar2(255 char))

**employee**

create table employee (employee\_type varchar2(31 char) not null, ssn number(19,0) not null, addreess\_line varchar2(255 char), city varchar2(255 char), country varchar2(255 char), name varchar2(255 char) not null, salary number(10,0) not null, state varchar2(255 char), zip\_code varchar2(255 char), store number(10,0), primary key (ssn))

**employee\_emails**

create table employee\_emails (employee\_id number(19,0) not null, emailid varchar2(255 char))

**employee\_manager**

create table employee\_manager (manager\_id number(19,0), employee\_id number(19,0) not null, primary key (employee\_id))

**employee\_phone\_numbers**

create table employee\_phone\_numbers (employee\_id number(19,0) not null, phone\_number number(10,0))

**store**

create table store (id number(10,0) not null, addreess\_line varchar2(255 char), city varchar2(255 char), country varchar2(255 char), name varchar2(255 char) not null, state varchar2(255 char), zip\_code varchar2(255 char), primary key (id))

**transaction**

create table transaction (id number(10,0) not null, category varchar2(255 char), charges double precision not null, delivery\_date date, delivery\_type varchar2(255 char), item\_type varchar2(255 char), start\_date date, weight varchar2(255 char), performed\_by number(19,0) not null, receiver number(10,0) not null, sender number(10,0) not null, store number(10,0) not null, primary key (id))

//TODO, add the constraints and index with meaningful names in the create statement only.

create index customer\_name\_idx on customer (name)

alter table customer\_emails add constraint FKiw8k6sl13asl9r77c4a8xtogb foreign key (customer\_id) references customer

alter table customer\_phone\_numbers add constraint FK8t59yk70tp1u41ltrlfkmk4ut foreign key (customer\_id) references customer

alter table employee add constraint FKofkajmv460crxhhp45s46rw3e foreign key (store) references store

alter table employee\_emails add constraint FKjatb5fwf4b6f1ch4b88kof4a3 foreign key (employee\_id) references employee

alter table employee\_manager add constraint FKmev5yh8c3pe4vqy6f4jsyhs46 foreign key (manager\_id) references employee

alter table employee\_manager add constraint FKa37g18wgt4dvi85px88kqsiat foreign key (employee\_id) references employee

alter table employee\_phone\_numbers add constraint FKi006k3mcv6oel47m5kmhnofsl foreign key (employee\_id) references employee

alter table transaction add constraint FK3ysgr89f4cw61nq5gbwinjpuv foreign key (performed\_by) references employee

alter table transaction add constraint FK890yl0i11cyfjt6bmvsxau55w foreign key (receiver) references customer

alter table transaction add constraint FK6351xnenfb2ik9usk8q7aixi6 foreign key (sender) references customer

alter table transaction add constraint FKmuibo0sm0ngbastj3a7m86pom foreign key (store) references store

**DML SQL**