**Table of Contents**

Page #

1. Requirement/Overview 2
2. ERD Model 4
3. Relational Schema 5
4. Data Dictionary 6
5. DDL 10
6. DML 10
7. Queries 14
8. Stored Procedures 18
9. Application Code 20
10. 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.

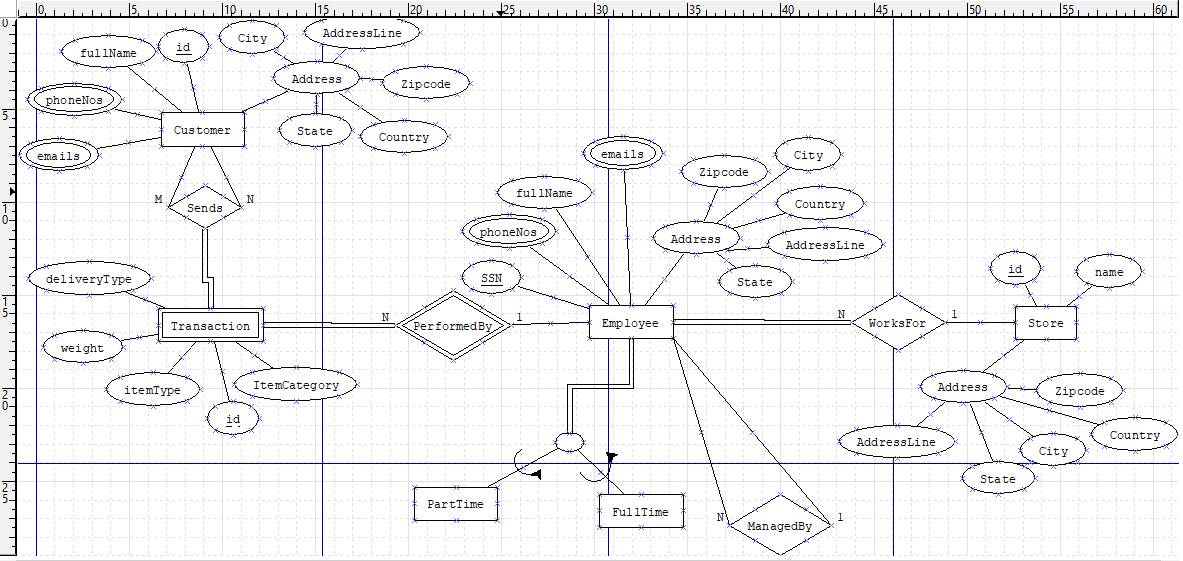
Manager: There is a hierarchy among the employees. An employee can report to a Manager, and Manager can have multiple employees working under him.

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)

customer\_phone\_numbers(id, phoneNos)

customer\_emails(id, emails)

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

Sends(CustomerId, TransactionId)

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

employee\_phone\_numbers(SSN, phoneNos)

employee\_emails(SSN, emails)

PartTime(EmployeeSSN)

FullTime(EmployeeSSN)

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

**Data Dictionary**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Customer**: Contains information about a Customer who wants to send a package using the postal management system. | | | | | | |
| **Attribute Name** | **Description** | **Datatype** | **Domain** | **Nullable** | **PK** | **FK** |
| ID | Customer Unique Identifier | NUMBER(10,0) | All | No | Yes | No |
| NAME | Customer Full Name | VARCHAR2(25) | All | No | No | No |
| ADDREESS\_LINE | Customer Street Address | VARCHAR2(255) | All | Yes | No | No |
| CITY | Customer City | VARCHAR2(50) | All | Yes | No | No |
| COUNTRY | Customer Country | VARCHAR2(50) | All | Yes | No | No |
| STATE | Customer State | VARCHAR2(50) | All | Yes | No | No |
| ZIP\_CODE | ZipCode of the area | VARCHAR2(50) | All | Yes | No | No |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Customer Email**: Contains information about Customer emails. | | | | | | |
| **Attribute Name** | **Description** | **Datatype** | **Domain** | **Nullable** | **PK** | **FK** |
| CUSTOMER\_ID | Customer Unique Identifier | NUMBER(10,0) | All | No | No | Yes |
| EMAILID | Customer Email ID | VARCHAR2(50) | All | Yes | No | No |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Customer Phone Numbers**: Contains information about Customer phone numbers. | | | | | | |
| **Attribute Name** | **Description** | **Datatype** | **Domain** | **Nullable** | **PK** | **FK** |
| CUSTOMER\_ID | Customer Unique Identifier | NUMBER(10,0) | All | No | No | Yes |
| PHONE\_NUMBER | Customer Phone number | NUMBER(10) | All | No | No | No |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Store**: Contains information about Stores that process the packages | | | | | | |
| **Attribute Name** | **Description** | **Datatype** | **Domain** | **Nullable** | **PK** | **FK** |
| ID | Store Unique Identifier | NUMBER(10,0) | All | No | Yes | No |
| NAME | Store name | VARCHAR2(100) | All | No | No | No |
| ADDREESS\_LINE | Store Street Address | VARCHAR2(255) | All | Yes | No | No |
| CITY | Store City | VARCHAR2(50) | All | Yes | No | No |
| COUNTRY | Store Country | VARCHAR2(50) | All | Yes | No | No |
| STATE | Store State | VARCHAR2(50) | All | Yes | No | No |
| ZIP\_CODE | ZipCode of the Store | VARCHAR2(50) | All | Yes | No | No |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Employee**: Contains information about Employees that process the packages | | | | | | |
| **Attribute Name** | **Description** | **Datatype** | **Domain** | **Nullable** | **PK** | **FK** |
| EMPLOYEE\_TYPE | Employee Type (Full time or Part time) | CHAR(4) | {“Full”, ”Part”} | No | No | No |
| SSN | Employee Unique identifier | CHAR(9) | {111-11-1111 – 999-99-9999} | No | Yes | No |
| NAME | Employee full name | VARCHAR2(100) | All | No | No | No |
| ADDREESS\_LINE | Employee Street Address | VARCHAR2(255) | All | Yes | No | No |
| CITY | Employee City | VARCHAR2(50) | All | Yes | No | No |
| COUNTRY | Employee Country | VARCHAR2(50) | All | Yes | No | No |
| STATE | Employee State | VARCHAR2(50) | All | Yes | No | No |
| ZIP\_CODE | ZipCode of the Employee | VARCHAR2(50) | All | Yes | No | No |
| STORE | Store unique identifier | NUMBER(10,0) | All | No | No | Yes |
| SALARY | Salary for full time employees | NUMBER(8,2) | All | Yes | No | No |
| HOURLY\_RATE | Hourly rate for part time employees | NUMBER(8,2) | All | Yes | No | No |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Employee Email**: Contains information about Employee emails. | | | | | | |
| **Attribute Name** | **Description** | **Datatype** | **Domain** | **Nullable** | **PK** | **FK** |
| EMPLOYEE\_SSN | Employee Unique Identifier | CHAR(9) | {111-11-1111 – 999-99-9999} | No | No | Yes |
| EMAILID | Employee Email ID | VARCHAR2(50) | All | Yes | No | No |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Employee Phone Numbers**: Contains information about Employee phone numbers. | | | | | | |
| **Attribute Name** | **Description** | **Datatype** | **Domain** | **Nullable** | **PK** | **FK** |
| EMPLOYEE\_SSN | Employee Unique Identifier | CHAR(9) | {111-11-1111 – 999-99-9999} | No | No | Yes |
| PHONE\_NUMBER | Employee Phone number | CHAR(11) | All | No | No | No |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Employee Manager**: Contains information about Employee Manager | | | | | | |
| **Attribute Name** | **Description** | **Datatype** | **Domain** | **Nullable** | **PK** | **FK** |
| MANAGER\_ID | Manager unique identifier | CHAR(9) | All | No | No | Yes |
| EMPLOYEE\_ID | Employee unique identifier | CHAR(9) | All | No | No | Yes |
| **Transaction**: Contains information about transactions made by the customer | | | | | | |
| **Attribute Name** | **Description** | **Datatype** | **Domain** | **Nullable** | **PK** | **FK** |
| ID | Transaction Unique Identifier | NUMBER(10,0) | All | No | Yes | No |
| CATEGORY | Item categorized based on size | VARCHAR2(10) | {'small','medium','large'} | No | No | No |
| CHARGES | Charges calculated for delivering the item | NUMBER(10,2) | All | No | No | No |
| DELIVERY\_DATE | Estimated date of delivery | DATE | All | Yes | No | No |
| DELIVERY\_TYPE | Type of delivery | VARCHAR2(10) | {'standard','overnight','urgent'} | No | No | No |
| ITEM\_TYPE | Type of Item | VARCHAR2(10) | {'delicate','normal','document'} | No | No | No |
| START\_DATE | Transaction start date | DATE | {SYSDATE} | No | No | No |
| WEIGHT | Weight of the package | NUMBER(4,3) | All | Yes | No | No |
| PERFORMED\_BY | Who performs the transaction | CHAR(9) | All | No | No | Yes |
| RECEIVER | The receiver of this item | NUMBER(10,0) | All | No | No | Yes |
| SENDER | The sender of this item | NUMBER(10,0) | All | No | No | Yes |

**DDL SQL**

**1.Customer**

create table customer (

id number(10,0) CONSTRAINT customer\_pk PRIMARY KEY,

name varchar2(25) CONSTRAINT customer\_uq\_name UNIQUE

CONSTRAINT customer\_nn\_name NOT NULL,

addreess\_line varchar2(255),

city varchar2(50),

country varchar2(50),

state varchar2(50),

zip\_code varchar2(50)

);

**2.Customer Email**

create table customer\_emails (

customer\_id number(10,0) CONSTRAINT customer\_nn\_id NOT NULL,

emailid varchar2(50),

CONSTRAINT customer\_fk\_email FOREIGN KEY (customer\_id) REFERENCES customer(id) ON DELETE CASCADE

);

**3. Customer phone numbers**

create table customer\_phone\_numbers (

customer\_id number(10,0) CONSTRAINT customer\_pno\_nn\_id NOT NULL,

phone\_number number(10) CONSTRAINT customer\_pno\_uq UNIQUE CONSTRAINT customer\_pno\_nn NOT NULL,

CONSTRAINT customer\_fk\_pno FOREIGN KEY (customer\_id) REFERENCES customer(id) ON DELETE CASCADE

);

**4. Store**

create table store (

id number(10,0) CONSTRAINT store\_pk PRIMARY KEY,

name varchar2(100) CONSTRAINT store\_nn\_name NOT NULL,

addreess\_line varchar2(255),

city varchar2(50),

country varchar2(50),

state varchar2(50),

zip\_code varchar2(50)

);

**5. Employee**

create table employee (

employee\_type char(4) CONSTRAINT employee\_nn\_type NOT NULL,

ssn char(9) CONSTRAINT employee\_pk PRIMARY KEY,

name varchar2(100) CONSTRAINT employee\_uq\_name UNIQUE

CONSTRAINT employee\_nn\_name NOT NULL,

addreess\_line varchar2(255),

city varchar2(50),

country varchar2(50),

state varchar2(50),

zip\_code varchar2(50),

store number(10,0),

salary number(8,2),

hourly\_rate number(8,2),

CONSTRAINT employee\_fk\_store FOREIGN KEY (store) REFERENCES store(id) ON DELETE SET NULL

);

**6. Employee email**

create table employee\_emails (

employee\_ssn char(9) CONSTRAINT employee\_email\_nn\_ssn NOT NULL,

emailid varchar2(50),

CONSTRAINT employee\_fk\_email FOREIGN KEY (employee\_ssn) REFERENCES employee(ssn) ON DELETE CASCADE

);

**7. Employee Phone number**

create table employee\_phone\_numbers (

employee\_ssn char(9) CONSTRAINT employee\_pno\_nn\_ssn NOT NULL,

phone\_number char(11) CONSTRAINT employee\_pno\_uq UNIQUE

CONSTRAINT employee\_pno\_nn NOT NULL,

CONSTRAINT employee\_fk\_pno FOREIGN KEY (employee\_ssn) REFERENCES employee(ssn) ON DELETE CASCADE

);

**8. Employee Manager**

create table employee\_manager (

manager\_id char(9) CONSTRAINT manager\_ssn\_nn NOT NULL ,

CONSTRAINT employee\_manager\_\_managerId\_fk FOREIGN KEY (manager\_id) REFERENCES employee(ssn) ON DELETE CASCADE,

employee\_id char(9) CONSTRAINT employee\_ssn\_nn NOT NULL,

CONSTRAINT employee\_manager\_\_empId\_fk FOREIGN KEY (employee\_id) REFERENCES employee(ssn) ON DELETE CASCADE

);

**9. Transaction**

create table transaction (

id number(10,0) CONSTRAINT transaction\_pk PRIMARY KEY,

category varchar2(10) CONSTRAINT transaction\_category\_chk check (category IN ('small','medium','large')),

charges number(10,2) CONSTRAINT transaction\_charges\_nn NOT NULL,

delivery\_date date ,

delivery\_type varchar2(10) CONSTRAINT transaction\_del\_type\_chk check (delivery\_type IN ('standard','overnight','urgent')),

item\_type varchar2(10) CONSTRAINT transaction\_type\_chk check (item\_type IN ('delicate','normal','document')),

start\_date date DEFAULT SYSDATE,

weight number(4,3),

performed\_by char(9) CONSTRAINT transaction\_employee\_nn NOT NULL, CONSTRAINT transaction\_fk\_employee FOREIGN KEY (performed\_by) REFERENCES employee(ssn) ON DELETE CASCADE,

receiver number(10,0) CONSTRAINT transaction\_receiver\_nn NOT NULL, CONSTRAINT transaction\_fk\_receiver FOREIGN KEY (receiver) REFERENCES customer(id) ON DELETE CASCADE,

sender number(10,0) CONSTRAINT transaction\_sender\_nn NOT NULL, CONSTRAINT transaction\_fk\_sender FOREIGN KEY (sender) REFERENCES customer(id) ON DELETE CASCADE

);

**DML:**

**Store:**

INSERT INTO "MANI\_DBA"."STORE" (ID, NAME, ADDREESS\_LINE, CITY, COUNTRY, STATE, ZIP\_CODE) VALUES ('1', 'UPS Store 1', 'Bethlhem Pik', 'Lansdale', 'USA', 'PA', '19446');INSERT INTO "MANI\_DBA"."STORE" (ID, NAME, ADDREESS\_LINE, CITY, COUNTRY, STATE, ZIP\_CODE) VALUES ('2', 'USA Northwales', 'Allen Town', 'Northwales', 'USA', 'PA', '19768');

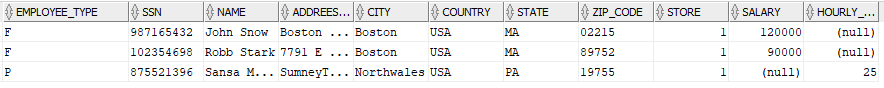


**Employee:**

INSERT INTO "MANI\_DBA"."EMPLOYEE" (EMPLOYEE\_TYPE, SSN, NAME, ADDREESS\_LINE, CITY, COUNTRY, STATE, ZIP\_CODE, STORE, SALARY) VALUES ('F', '987165432', 'John Snow', 'Boston MA. 1185 Boylston St.', 'Boston', 'USA', 'MA', '02215', '1', '120000');

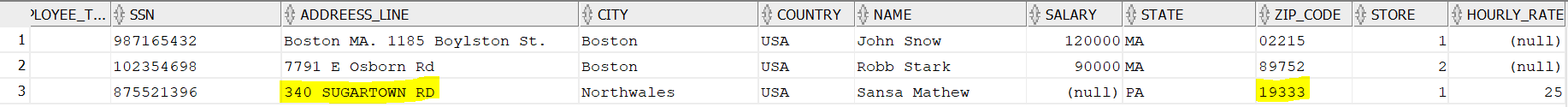
INSERT INTO "MANI\_DBA"."EMPLOYEE" (EMPLOYEE\_TYPE, SSN, NAME, ADDREESS\_LINE, CITY, COUNTRY, STATE, ZIP\_CODE, STORE, SALARY) VALUES ('F', '102354698', 'Robb Stark', '7791 E Osborn Rd', 'Boston', 'USA', 'MA', '89752', '1', '90000');

INSERT INTO "MANI\_DBA"."EMPLOYEE" (EMPLOYEE\_TYPE, SSN, NAME, ADDREESS\_LINE, CITY, COUNTRY, STATE, ZIP\_CODE, STORE, HOURLY\_RATE) VALUES ('P', '875521396', 'Sansa Mathew', 'SumneyTown Pike Rd.', 'Northwales', 'USA', 'PA', '19755', '1', '25');



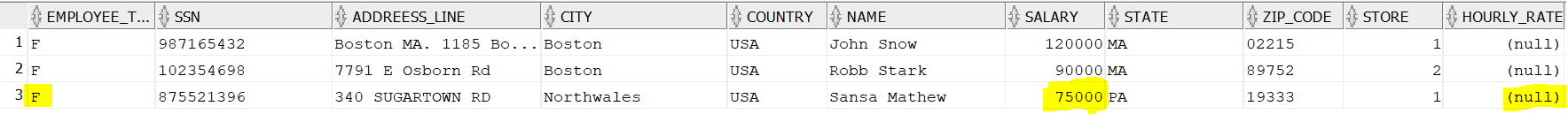
UPDATE "SVEMUGUN"."EMPLOYEE" SET ADDREESS\_LINE='340 SUGARTOWN RD', ZIP\_CODE='19333' WHERE SSN='875521396';

COMMIT;



UPDATE "SVEMUGUN"."EMPLOYEE" SET EMPLOYEE\_TYPE='F', SALARY='75000', HOURLY\_RATE=NULL WHERE SSN='875521396';

COMMIT;

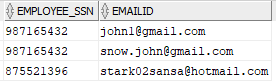


**Employee\_emails:**

INSERT INTO "MANI\_DBA"."EMPLOYEE\_EMAILS" (EMPLOYEE\_SSN, EMAILID) VALUES ('987165432', 'john1@gmail.com');

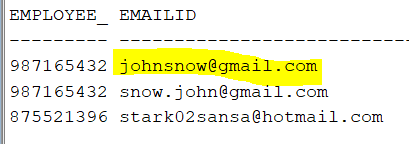
INSERT INTO "MANI\_DBA"."EMPLOYEE\_EMAILS" (EMPLOYEE\_SSN, EMAILID) VALUES ('987165432', 'snow.john@gmail.com');

INSERT INTO "MANI\_DBA"."EMPLOYEE\_EMAILS" (EMPLOYEE\_SSN, EMAILID) VALUES ('875521396', 'stark02sansa@hotmail.com');



UPDATE "SVEMUGUN"."EMPLOYEE\_EMAILS" SET EMAILID='johnsnow@gmail.com' WHERE EMPLOYEE\_SSN='987165432' AND EMAILID='john1@gmail.com';

COMMIT;

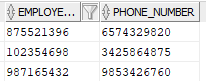


**Employee\_phonenumbers:**

INSERT INTO "MANI\_DBA"."EMPLOYEE\_PHONE\_NUMBERS" (EMPLOYEE\_SSN, PHONE\_NUMBER) VALUES ('875521396', '6574329820');

INSERT INTO "MANI\_DBA"."EMPLOYEE\_PHONE\_NUMBERS" (EMPLOYEE\_SSN, PHONE\_NUMBER) VALUES ('102354698', '3425864875');

INSERT INTO "MANI\_DBA"."EMPLOYEE\_PHONE\_NUMBERS" (EMPLOYEE\_SSN, PHONE\_NUMBER) VALUES ('987165432', '9853426760');



DELETE FROM EMPLOYEE\_PHONE\_NUMBER

**Employee\_Manager:**

INSERT INTO "MANI\_DBA"."EMPLOYEE\_MANAGER" (MANAGER\_ID, EMPLOYEE\_ID) VALUES ('987165432', '102354698');

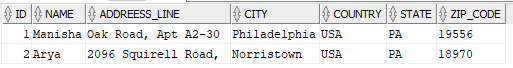
INSERT INTO "MANI\_DBA"."EMPLOYEE\_MANAGER" (MANAGER\_ID, EMPLOYEE\_ID) VALUES ('987165432', '875521396');



**Customer:**

INSERT INTO "MANI\_DBA"."CUSTOMER" (ID, NAME, ADDREESS\_LINE, CITY, COUNTRY, STATE, ZIP\_CODE) VALUES ('1', 'Manisha', 'Oak Road, Apt A2-30', 'Philadelphia', 'USA', 'PA', '19556');

INSERT INTO "MANI\_DBA"."CUSTOMER" (ID, NAME, ADDREESS\_LINE, CITY, COUNTRY, STATE, ZIP\_CODE) VALUES ('2', 'Arya', '2096 Squirell Road, ', 'Norristown', 'USA', 'PA', '18970');

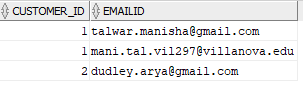


**Customer\_emails:**

INSERT INTO "MANI\_DBA"."CUSTOMER\_EMAILS" (CUSTOMER\_ID, EMAILID) VALUES ('1', 'talwar.manisha@gmail.com');

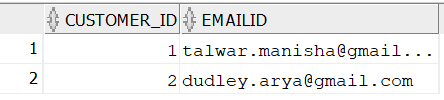
INSERT INTO "MANI\_DBA"."CUSTOMER\_EMAILS" (CUSTOMER\_ID, EMAILID) VALUES ('1', 'mani.tal.vil297@villanova.edu');

INSERT INTO "MANI\_DBA"."CUSTOMER\_EMAILS" (CUSTOMER\_ID, EMAILID) VALUES ('2', 'dudley.arya@gmail.com');



DELETE FROM "SVEMUGUN"."CUSTOMER\_EMAILS" WHERE CUSTOMER\_ID='1' AND EMAILID='mani.tal.vil297@villanova.edu';

COMMIT;

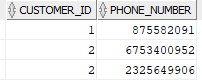


**Customer\_phonenumbers:**

INSERT INTO "MANI\_DBA"."CUSTOMER\_PHONE\_NUMBERS" (CUSTOMER\_ID, PHONE\_NUMBER) VALUES ('1', '875582091');

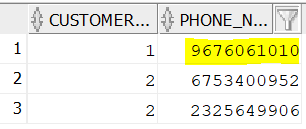
INSERT INTO "MANI\_DBA"."CUSTOMER\_PHONE\_NUMBERS" (CUSTOMER\_ID, PHONE\_NUMBER) VALUES ('2', '6753400952');

INSERT INTO "MANI\_DBA"."CUSTOMER\_PHONE\_NUMBERS" (CUSTOMER\_ID, PHONE\_NUMBER) VALUES ('2', '2325649906');



UPDATE "SVEMUGUN"."CUSTOMER\_PHONE\_NUMBERS" SET PHONE\_NUMBER='9676061010' WHERE CUSTOMER\_ID='1';

COMMIT;



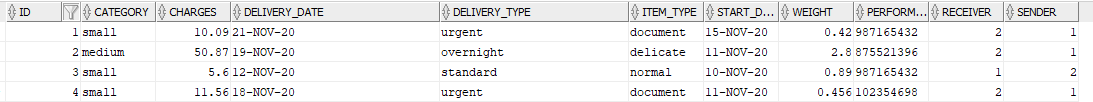
**Transaction:**

INSERT INTO "MANI\_DBA"."TRANSACTION" (ID, CATEGORY, CHARGES, DELIVERY\_TYPE, ITEM\_TYPE, WEIGHT, PERFORMED\_BY, RECEIVER, SENDER) VALUES ('1', 'small', '10.09', 'urgent', 'document', '0.420', '987165432', '2', '1');

INSERT INTO "MANI\_DBA"."TRANSACTION" (ID, CATEGORY, CHARGES, DELIVERY\_TYPE, ITEM\_TYPE, WEIGHT, PERFORMED\_BY, RECEIVER, SENDER) VALUES ('2', 'medium', '50.87', 'overnight', 'delicate', '2.800', '875521396', '2', '1');

INSERT INTO "MANI\_DBA"."TRANSACTION" (ID, CATEGORY, CHARGES, DELIVERY\_TYPE, ITEM\_TYPE, WEIGHT, PERFORMED\_BY, RECEIVER, SENDER) VALUES ('3', 'small', '5.60', 'standard', 'normal', '0.890', '987165432', '1', '2');

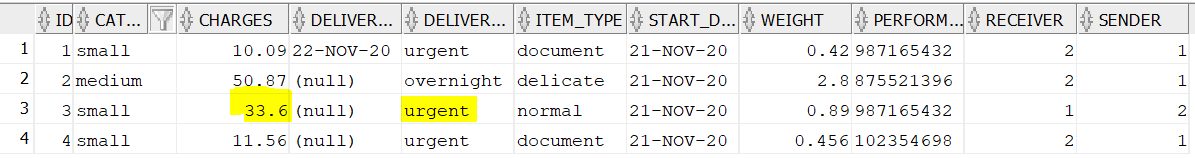
INSERT INTO "MANI\_DBA"."TRANSACTION" (ID, CATEGORY, CHARGES, DELIVERY\_TYPE, ITEM\_TYPE, WEIGHT, PERFORMED\_BY, RECEIVER, SENDER) VALUES ('4', 'small', '11.56', 'urgent', 'document', '0.456', '102354698', '2', '1');



UPDATE "MANI\_DBA"."TRANSACTION" SET DELIVERY\_DATE =to\_date('22-Nov-20') WHERE id=1;

UPDATE "SVEMUGUN"."TRANSACTION" SET DELIVERY\_TYPE ='urgent', CHARGES=CHARGES\*3 WHERE id=3;

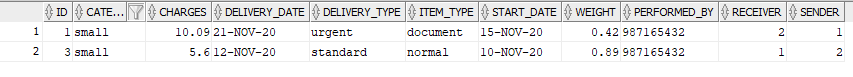
COMMIT;



**Queries**

1. Find recent transactions details performed by an employee in a store.

Select \* FROM transaction t WHERE performed\_by= 987165432 ORDER BY delivery\_date DESC;



1. Find all full time employees name and salary working for a store whose salary is > 95000

Select name, salary

FROM employee

WHERE employee\_type='F' AND store=1 AND salary > 95000;



1. Display all employees name and status working under a particular manager in a store.

Select e.name, e.employee\_type

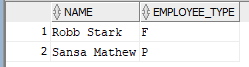
FROM employee e

WHERE e.ssn IN (

Select em.employee\_id FROM employee\_manager em

WHERE em.manager\_id= 987165432

) AND e.store=1

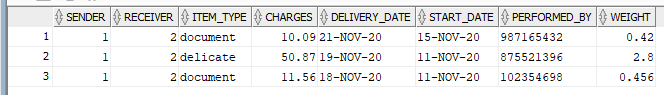


1. Find all the transactions performed between a date range.

Select t.sender, t.receiver, t.item\_type, t.charges, t.delivery\_date, t.start\_date, t.performed\_by,t.weight

FROM transaction t

WHERE t.start\_date BETWEEN to\_date('11-Nov-20') AND to\_date('18-Nov-20');

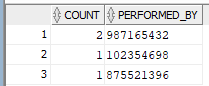


1. Find number of transactions performed by each employee.

Select COUNT(t.sender) as count, t.performed\_by

FROM transaction t

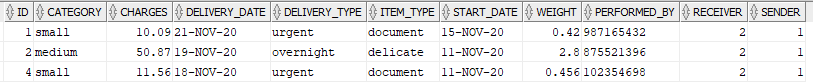
GROUP BY t.performed\_by



1. Find the transaction that are left to be delivered.

Select \* FROM transaction t

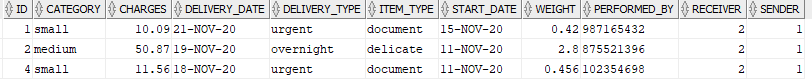
WHERE t.delivery\_date > SYSDATE + 1;



1. Find all the urgent and overnight deliveries requested by a customer/sender.

Select \* FROM transaction t

WHERE t.delivery\_type IN ('urgent', 'overnight') AND sender=1;



1. How many employees are working full time/part time?

Select e.employee\_type, count(e.ssn) AS count FROM employee e GROUP BY e.employee\_type;

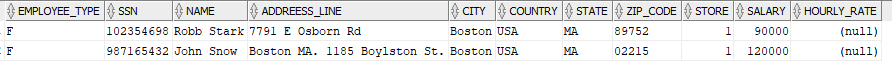


1. Top 2 employees who delivers the more number of packages

SELECT \* FROM employee where ssn in

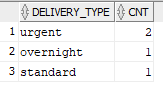
(select performed\_by from

(select performed\_by,count(id) AS cnt FROM transaction GROUP BY performed\_by order by cnt desc) where rownum<=2);



1. Display number of transactions for each delivery type and what is the most frequent delivery type customers choose?

SELECT delivery\_type,count(id) AS cnt FROM transaction GROUP BY delivery\_type ORDER BY cnt DESC;



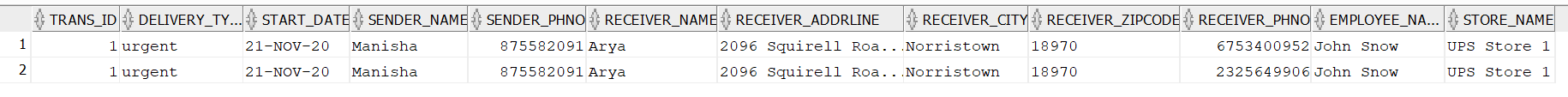
SELECT delivery\_type FROM (SELECT delivery\_type,count(id) AS cnt FROM transaction GROUP BY delivery\_type ORDER BY cnt DESC)

WHERE rownum=1;



1. Fetch the transactions, customer and employee details of packages on a particular date:

SELECT trans.id AS trans\_id, trans.delivery\_type,trans.start\_date, cust\_sender.name AS sender\_name, cust\_sender\_phno.phone\_number AS sender\_phno, cust\_receiver.name AS receiver\_name,cust\_receiver.addreess\_line AS receiver\_addrline ,cust\_receiver.city AS receiver\_city,cust\_receiver.zip\_code AS receiver\_zipcode,cust\_receiver\_phno.phone\_number AS receiver\_phno, emp.name AS employee\_name,str.name AS store\_name FROM transaction trans,customer cust\_sender,customer cust\_receiver,employee emp, store str, customer\_phone\_numbers cust\_sender\_phno,customer\_phone\_numbers cust\_receiver\_phno WHERE trans.performed\_by=emp.ssn AND trans.sender=cust\_sender.id AND trans.receiver=cust\_receiver.id AND emp.store=str.id AND cust\_sender.id=cust\_sender\_phno.customer\_id AND cust\_receiver.id=cust\_receiver\_phno.customer\_id AND trans.delivery\_date=to\_date(' 22-Nov-20');



**Stored Procedure:**

1. Find all the transactions where delivery type is ‘urgent’ and item\_type is ‘document’, and double their delivery charges.

Procedure Definition:

*CREATE OR REPLACE PROCEDURE updateTransCharges(*

*p\_delivery\_type IN TRANSACTION.DELIVERY\_TYPE%TYPE,*

*p\_item\_type IN TRANSACTION.ITEM\_TYPE%TYPE)*

*IS*

*BEGIN*

*UPDATE TRANSACTION SET CHARGES = CHARGES \* 2.0 WHERE DELIVERY\_TYPE = p\_delivery\_type AND ITEM\_TYPE = p\_item\_type;*

*COMMIT;*

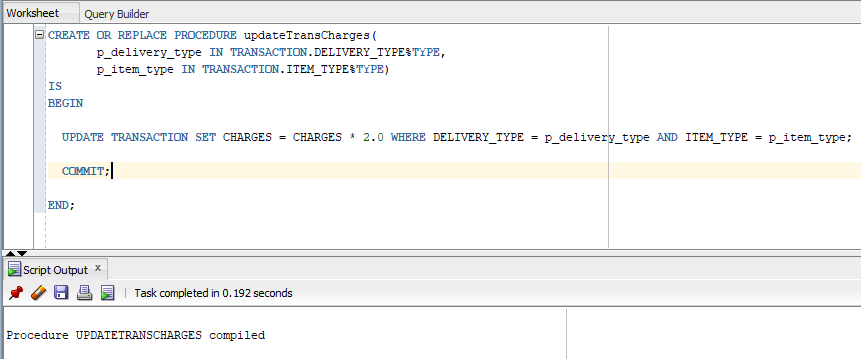
*END;*

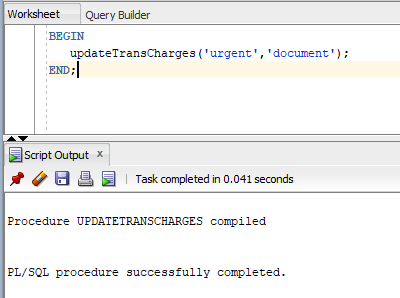
Procedure invocation:

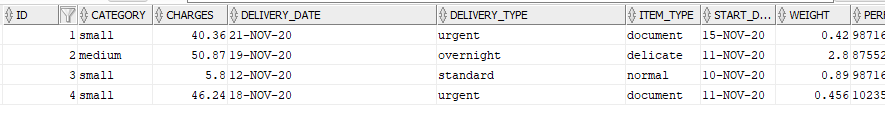
*BEGIN*

*updateTransCharges('urgent','document');*

*END;*







1. Update the work location for an employee

Procedure Definition:

*CREATE OR REPLACE PROCEDURE updateEmpWorkLocation(*

*p\_SSN IN EMPLOYEE.SSN%TYPE,*

*p\_STORE\_ID IN EMPLOYEE.STORE%TYPE)*

*IS*

*EMP\_NAME VARCHAR2(100);*

*BEGIN*

*SELECT NAME into EMP\_NAME FROM EMPLOYEE WHERE SSN=p\_SSN;*

*UPDATE EMPLOYEE SET STORE = p\_STORE\_ID WHERE SSN = p\_SSN;*

*COMMIT;*

*DBMS\_OUTPUT.PUT\_LINE('The updated work location of the employee ' || EMP\_NAME ||' is ' || p\_STORE\_ID);*

*END;*

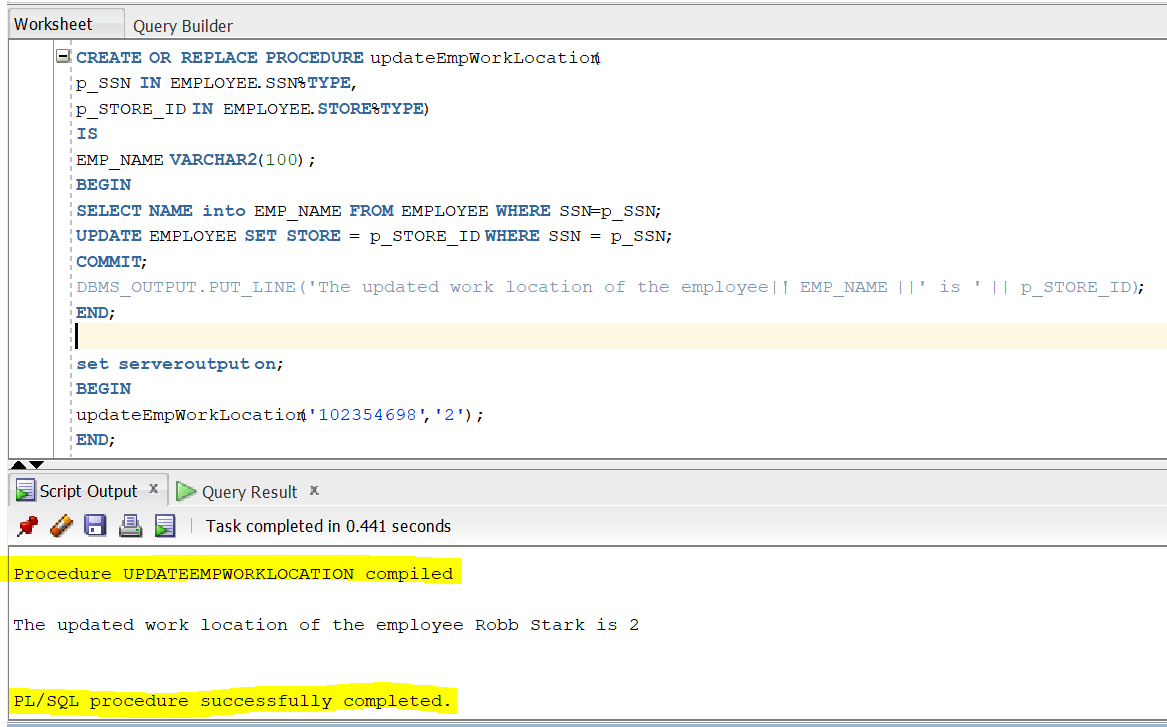
Procedure invocation:

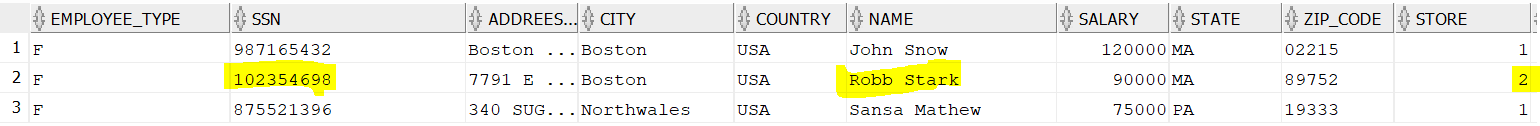
*set serveroutput on;*

*BEGIN*

*updateEmpWorkLocation('102354698','2');*

*END;*





**Application Code**

Git-Hub: <https://github.com/TmanishaT/CS8490-Database-Systems>

Connection properties:

spring.jpa.generate-ddl=true

spring.jpa.hibernate.ddl-auto=update

spring.datasource.url=jdbc:oracle:thin:@45.79.135.253:1521/xe

spring.datasource.username=mtanwar

spring.datasource.password=02205833

spring.datasource.driver-class-name=oracle.jdbc.OracleDriver

## this shows the sql actions in the terminal logs

spring.jpa.show-sql=true

Repositories:

**Transaction Repository**

@Repository

**public** **interface** TransactionRepository **extends** JpaRepository<Transaction, Integer>{

@Query("SELECT t FROM Transaction t WHERE t.sender=:senderId")

**public** List<Transaction> findAllBySenderId(@Param("senderId") Integer senderId);

@Query("SELECT t FROM Transaction t WHERE t.receiver=:receiverId")

**public** List<Transaction> findAllByReceiverId(@Param("receiverId") Integer receiverId);

@Query("SELECT t FROM Transaction t WHERE t.itemType=:itemType")

**public** List<Transaction> findAllByItemType(@Param("itemType") String itemType);

@Query("SELECT t FROM Transaction t WHERE t.startDate=:startDate")

**public** List<Transaction> findAllByStartDate(@Param("startDate") Date startDate);

@Query("SELECT t FROM Transaction t WHERE t.deliveryDate=:deliveryDate")

**public** List<Transaction> findAllByDeliveryDate(@Param("deliveryDate") Date deliveryDate);

@Query("SELECT t FROM Transaction t WHERE t.deliveryType=:deliveryType")

**public** List<Transaction> findAllByDeliveryType(@Param("deliveryType") String deliveryType);

@Query("SELECT t FROM Transaction t WHERE t.category=:itemCategory")

**public** List<Transaction> findAllByItemCategory(@Param("itemCategory") String itemCategory);

@Query("SELECT t FROM Transaction t WHERE t.store=:storeId")

**public** List<Transaction> findAllByStoreId(@Param("storeId") Integer storeId);

@Query("SELECT t FROM Transaction t WHERE t.performedBy=:performedBy")

**public** List<Transaction> findAllByManagerId(@Param("performedBy") Integer performedBy);

}

**Store Repository**

@Repository

**public** **interface** StoreRepository **extends** JpaRepository<Store, Integer> {

@Query("SELECT s FROM Store s WHERE s.name=:storeName")

List<Store> findByName(String storeName);

}

**Employee Repository**

@Repository

**public** **interface** EmployeeRepository **extends** JpaRepository<Employee, Long> {

@Query("SELECT e FROM Employee e WHERE e.name=:name")

List<Employee> findByName(String name);

@Query("SELECT e FROM Employee e WHERE e.store=:storeId")

List<Employee> findByStoreId(Integer storeId);

@Query("SELECT e FROM Employee e WHERE e.SSN IN (:employeeIds)")

List<Employee> findBySSNList(List<Long> employeeIds);

}

**Customer Repository**

@Repository

**public** **interface** CustomerRepository **extends** JpaRepository<Customer, Integer>{

@Query

("SELECT c FROM Customer c WHERE LOWER(c.name) = LOWER(:custName)")

**public** List<Customer> findByCustomerName(@Param("custName") String custName);

}