Post Office Database

Bachelor of Technology
Computer Science and Engineering

Submitted By

KAUSHIK GHOSH (13000116101)
AMAN KUMAR MISHRA(13000116137)
PULKIT(13000116081)
YASSAR ALI(13000116022)

MAY 2019

Techno India
EM-4/1, Sector-V, Salt Lake
Kolkata- 700091
West Bengal
India

TABLE OF CONTENTS

1. Abstract

Post Office since the beginning has served as one of the pioneers in developing the connections in between the communities staying far away. In any country, the information that reaches from one point to other, plays a very important role, from letters to couriers, to money management, before the commercial banks took over, Post Offices has been serving as the best medium to be connected to the world from east to the west and from north to the south. Thus Post Office, with its growing popularity and trust, needed a trustworthy data handling model, that could take care of all the scrambled data coming in with the help of modern day technologies in this case Database Management Systems. This paper takes care of all the possibilities and requirements of a post office at this present day, and is updated and scalable for real world use.

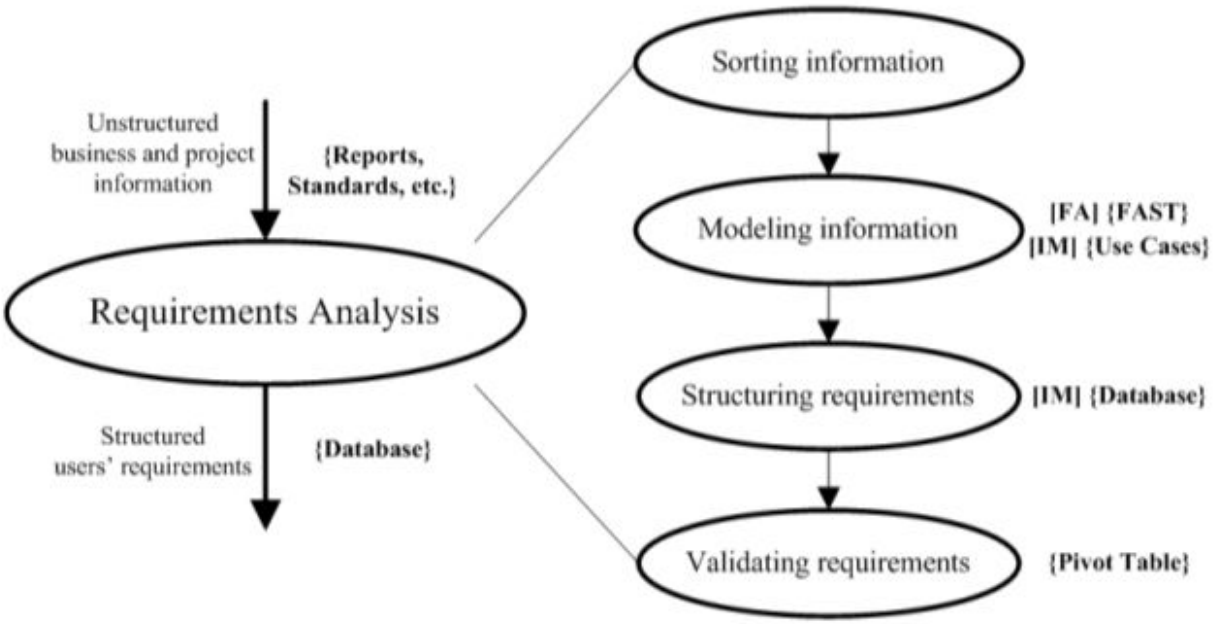
2. Introduction

As Discussed in the abstract, to support such a large database, It should be split into smaller tables which can be maintained easily without any hassle. The paper will be first discussing the requirements of the managements, ie the various entities that are involved in it as well as the relationships that they have in between themselves, termed as Requirement Analysis. After this, It will move towards the ER(Entity-Relationship) Diagram which is a pictorial view of the model that we're trying to design. The relational schema will then show how the entities are stored in the table. Since, in real world there will be a lot of data, so to consume the least time, and money optimization at the basic level also needs to be done, normalisation using BCNF decomposition along with dependency preserving takes care of this. After this to further demonstrate the utilities of this model on the basis of some assumptions, some queries are executed, which further shows the scalability of the model in the real world.

3. Body

3.i Requirement Analysis

Requirements Analysis is the process of defining the expectations of the users for an application that is to be built or modified. Requirements analysis involves all the tasks that are conducted to identify the needs of different stakeholders. Therefore requirements analysis means to analyze, document, validate and manage software or system requirements. High-quality requirements are documented, actionable, measurable, testable, traceable, helps to identify business opportunities, and are defined to a facilitate system design.



3.i.a Unstructured Information

The Post Office database has the following business needs

a. A table to store the employee details, such as names, addresses, phone numbers, date of birth, age, designation and salary
b. A table to organise the customers database, the name, address, whether he is a sender or a receiver, aadhar id, phone number and email id.
c. A table to take care of the daily transactions for example, a letter being sent to someone, or a withdrawal from the account.
d. A table to take care of the services that any post office has to offer, for example not every post office can offer monetary banking services or speed post services.

The Post Office has the following technical needs

a. To Scale a model so big, Microsoft Excel or any table filling software won't serve the purpose.
b. The relationships that can be served on the entities, can't be correlated just with the use of simple table filling softwares.

3.i.b Solutions

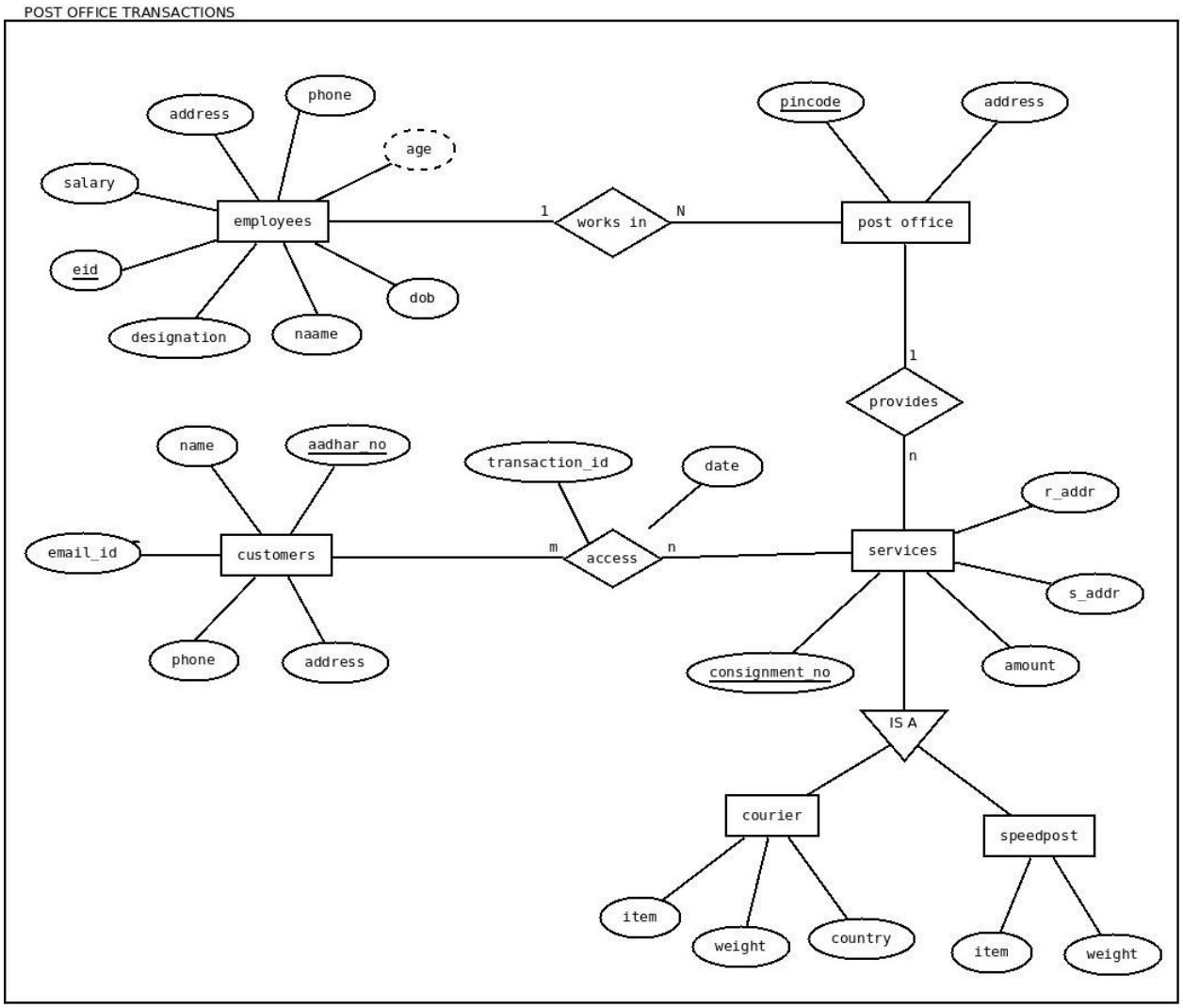
a. A database using oracle 10g as it is scalable for businesses, and also easy to maintain, as oracle continuously provides services for all its products.
b. The ER Diagram using Dia, which servers as a design model for the databases, and its relational schema.
c. The model is also then normalized using BCNF decomposition, and dependency preserving.
d. The queries are then tested on the database, to see how the data and the relationships are performing.

3.i.c Entities derived from requirements

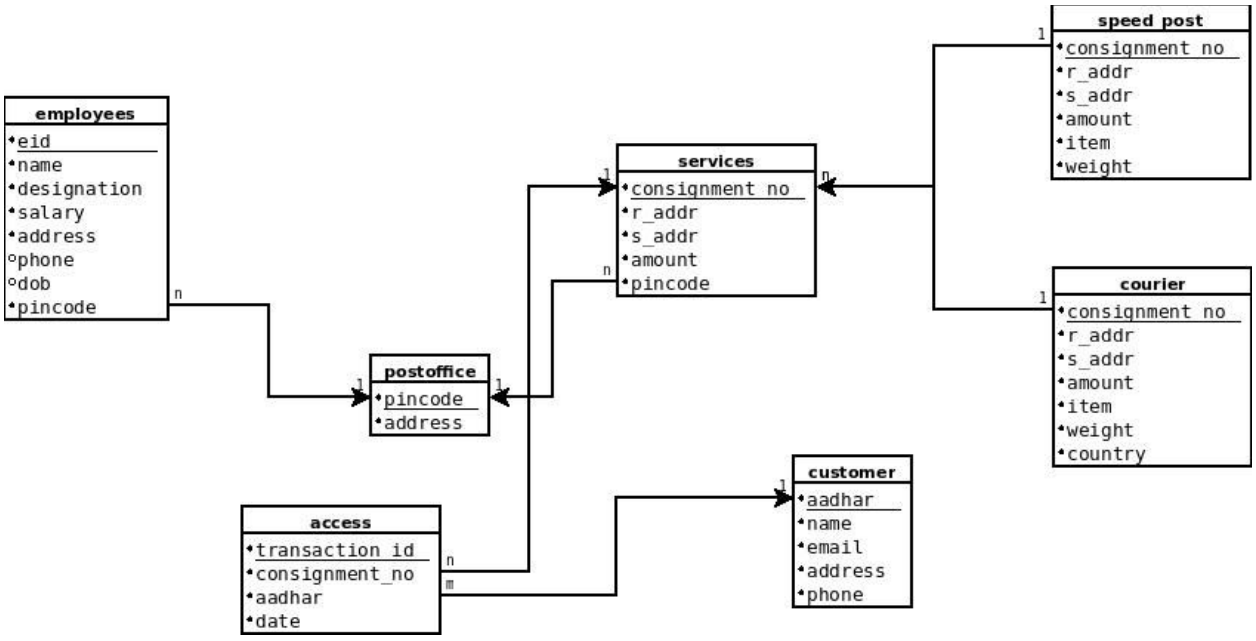
Employees	Post office needs to store all the details of employees which are working for the post office, for which it needs the entity type EMPLOYEE. EMPLOYEE is a generalized entity because there are different types of employees(e.g Postman,Clerk,Ground Staff,Manager).
Post Office	A post office is a public department that provides a customer service to the public and handles their mail needs.Post offices offer mail-related services such as acceptance of letters and parcels; provision of post office boxes; and sale of postage stamps, packaging, and stationery
Services	Post office provides different kinds of services to the customers .Services for saving scheme and even mail services. SERVICES is a generalised entity because there are different types of SERVICES (e.g Courier, Speed Post etc).
Courier	Courier service is provided by any Post Office for the customers to transfer important informations and packages from one place to another. In our design Courier is a sub part of Services.
Speed Post	The speed post transmission follows a hub and spoke model. All the pincode in India are mapped to one of the hubs. The post office will send the shipment to its parent hub. Speed Post service is provided by post office and in our project it is a subpart of Services.
Customer	A customer is an individual or business that purchases another company's goods or services. So in case of Post Office Database Management System we keep the records of all the customers who come to the post office for either receiving or sending telegrams or for any other works.

3.ii ER MODEL

An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.



3.iii Relational Schema



3.iv Assumptions

The following are the assumptions associated with this design.

-> Employees to post office is a 1 to many relationship.

-> post office to services is 1 to many relationship.

-> post office to services is 1 to many relationship.

-> services are of two types:- a)courier b)speed post.

-> customers to services is many to many relationship.

-> The relation access is modeled as aggregation namely postoffice transactions.

3.v Normalization

Functional Dependencies of the entities
Employees(eid,pincode,name,designation,address,phone,dob,salary)

1. Employees

   eid,pincode,phonenumber -> name,designation,salary,address,phone,dob,pincode
   eid,dob -> name,designation,salary,address

Determining highest normal form of relation:

Input relation is assumed to be in 1NF: each attribute is assumed to contain only one value per row.

Input relation is in 2NF: It is in 1NF and there are no partial dependencies on a composite minimum key (a minimum key composed of more than one attribute).

Input relation is not in 3NF: it is in 2NF but not all functional dependencies satisfy at least one of the following conditions: (1) The right-hand side is a subset of the left hand side, (2) the left-hand side is a superkey (or minimum key) of the relation, or (3) the right-hand side is (or is a part of) some minimum key of the relation. The functional dependencies that failed are: DOB,EID → NAME; DOB,EID → SALARY; DOB,EID → ADDRESS; DOB,EID → DESIGNATION.

Input relation is not in BCNF: it is not in 3NF and not all functional dependencies satisfy at least one of the following conditions: (1) The right-hand side is a subset of the left hand side, or (2) the left-hand side is a superkey (or minimum key) of the relation. The functional dependencies that failed are: DOB,EID → NAME; DOB,EID → SALARY; DOB,EID → ADDRESS; DOB,EID → DESIGNATION.

Input relation is not in 4NF: it is not in BCNF and not all nontrivial multivalued dependencies satisfied the 4NF condition that the left-hand side is a superkey (or minimum key) of the relation. A multivalued dependency is trivial if either (1) the right-hand side is a subset of the left-hand side, or (2) the multivalued dependency contains all attributes of the input relation. The multivalued dependencies that failed are: DOB,EID ↠ NAME; DOB,EID ↠ SALARY; DOB,EID ↠ ADDRESS; DOB,EID ↠ DESIGNATION.

Decomposing input relation into 3NF using canonical functional dependency cover (lossless and preserving all minimal cover set functional dependencies):

Decomposing input relation into 3NF relations using the Synthesis algorithm. For each functional dependency of the canonical cover set (merging functional dependencies having the same left-hand attribute(s)) of original relation's functional dependencies, create a relation schema with the attributes in that functional dependency (both sides). Checking if at least one key can be found in at least one newly formed 3NF relation. Since key {EID, PHONE, PINCODE} is present in at least one of the new 3NF relations, no new relation was created. Testing if any relation includes all of the attributes found in another relation (and deleting the duplicate or smaller one). No new relations were removed. Finished decomposing input relation into 3NF relations:

EMPLOYEES0(DOB, EID, NAME, SALARY, ADDRESS, DESIGNATION) having FD(s): DOB,EID → NAME; DOB,EID → SALARY; DOB,EID → ADDRESS; DOB,EID → DESIGNATION.

EMPLOYEES1(DOB, EID, PHONE, PINCODE) having FD(s): EID,PHONE,PINCODE → DOB.

Decomposing input relation into BCNF relations (lossless but not necessarily functional dependency preserving). Will attempt two parallel decompositions: one from the input relation and the second from the set of decomposed 3NF relations:

Decomposing input relation into BCNF relations using input relation and input functional dependencies as sources.

Final set of decomposed BCNF relations:

EMPLOYEES_0(DOB, EID, NAME, SALARY, ADDRESS, DESIGNATION) having FD(s): DOB,EID → NAME; DOB,EID → SALARY; DOB,EID → ADDRESS; DOB,EID → DESIGNATION.

EMPLOYEES_1(DOB, EID, PHONE, PINCODE) having FD(s): EID,PHONE,PINCODE → DOB; EID,PHONE,PINCODE → PHONE; EID,PHONE,PINCODE → PINCODE.

The following input functional dependency was lost: EID,PHONE,PINCODE → DOB,NAME,PHONE,SALARY,ADDRESS,PINCODE,DESIGNATION.

Note that a lost input functional dependency can be safely ignored if it is not part of the minimal cover set of functional dependencies

Decomposing input relation into BCNF relations using decomposed 3NF relations and the minimal cover set of functional dependencies as sources.

Final set of decomposed BCNF relations:

EMPLOYEES0(DOB, EID, NAME, SALARY, ADDRESS, DESIGNATION) having FD(s): DOB,EID → NAME; DOB,EID → SALARY; DOB,EID → ADDRESS; DOB,EID → DESIGNATION.

EMPLOYEES1(DOB, EID, PHONE, PINCODE) having FD(s): EID,PHONE,PINCODE → DOB.

No functional dependencies from the minimal cover set were lost.

2.Services

services(consignment_no,r_addr,s_addr,amount,item,weight)
consignment_no -> r_addr,s_addr,amount,pincode
pincode -> consignment_no,r_addr,s_addr,pincode

Determining highest normal form of relation:

Input relation is assumed to be in 1NF: each attribute is assumed to contain only one value per row.

Input relation is not in 2NF: There is at least one partial dependency on a composite minimum key. To satisfy 2NF, there should not be any non-prime attribute that can be functionally determined by a proper subset of a composite minimum key. See above closure list for the composite minimum key(s). The following non-prime attributes violate the condition: The minimum set of attributes that attribute RADDR is functionally determined by is attribute(s) {CONSIGNMENTNO}; whereas it should only be functionally determined by the full set of attributes of the composite minimum key {ITEM, WEIGHT, CONSIGNMENTNO}. The minimum set of attributes that attribute SADDR is functionally determined by is attribute(s) {CONSIGNMENTNO}; whereas it should only be functionally determined by the full set of attributes of the composite minimum key {ITEM, WEIGHT, CONSIGNMENTNO}. The minimum set of attributes that attribute AMOUNT is functionally determined by is attribute(s) {CONSIGNMENTNO}; whereas it should only be functionally determined by the full set of attributes of the composite minimum key {ITEM, WEIGHT, CONSIGNMENTNO}.

Input relation is not in 3NF: it is not in 2NF and not all functional dependencies satisfy at least one of the following conditions: (1) The right-hand side is a subset of the left hand side, (2) the left-hand side is a superkey (or minimum key) of the relation, or (3) the right-hand side is (or is a part of) some minimum key of the relation. The functional dependencies that failed are: CONSIGNMENTNO → AMOUNT; PINCODE → RADDR; PINCODE → SADDR.

Input relation is not in BCNF: it is not in 3NF and not all functional dependencies satisfy at least one of the following conditions: (1) The right-hand side is a subset of the left hand side, or (2) the left-hand side is a superkey (or minimum key) of the relation. The functional dependencies that failed are: CONSIGNMENTNO → AMOUNT; CONSIGNMENTNO → PINCODE; PINCODE → RADDR; PINCODE → SADDR; PINCODE → CONSIGNMENTNO.

Input relation is not in 4NF: it is not in BCNF and not all nontrivial multivalued dependencies satisfied the 4NF condition that the left-hand side is a superkey (or minimum key) of the relation. A multivalued dependency is trivial if either (1) the right-hand side is a subset of the left-hand side, or (2) the multivalued dependency contains all attributes of the input relation. The multivalued dependencies that failed are: CONSIGNMENTNO ↠ AMOUNT; CONSIGNMENTNO ↠ PINCODE; PINCODE ↠ RADDR; PINCODE ↠ SADDR; PINCODE ↠ CONSIGNMENTNO.

Decomposing input relation into 3NF using canonical functional dependency cover (lossless and preserving all minimal cover set functional dependencies):

Decomposing input relation into 3NF relations using the Synthesis algorithm. For each functional dependency of the canonical cover set (merging functional dependencies having the same left-hand attribute(s)) of original relation's functional dependencies, create a relation schema with the attributes in that functional dependency (both sides). There is at least one attribute from original relation that has not been placed in any new relation, so creating additional relation schema with those attribute(s): ITEM WEIGHT. Checking if at least one key can be found in at least one newly formed 3NF relation. Since none of the newly created 3NF relations contains a key of the original relation, need to add another relation whose schema is a key of the original relation. Added key {ITEM, WEIGHT, CONSIGNMENTNO}. Testing if any relation includes all of the attributes found in another relation (and deleting the duplicate or smaller one). Removed at least one new relation that was a duplicate or subset of another new relation. Finished decomposing input relation into 3NF relations:

SERVICES0(AMOUNT, PINCODE, CONSIGNMENTNO) having FD(s): CONSIGNMENTNO → AMOUNT; CONSIGNMENTNO → PINCODE; PINCODE → CONSIGNMENTNO.

SERVICES1(RADDR, SADDR, PINCODE, CONSIGNMENTNO) having FD(s): CONSIGNMENTNO → PINCODE; PINCODE → RADDR; PINCODE → SADDR; PINCODE → CONSIGNMENTNO.

SERVICES3(ITEM, WEIGHT, CONSIGNMENTNO) having FD(s): (none).

Decomposing input relation into BCNF relations (lossless but not necessarily functional dependency preserving). Will attempt two parallel decompositions: one from the input relation and the second from the set of decomposed 3NF relations:

Decomposing input relation into BCNF relations using input relation and input functional dependencies as sources.

Initial set of decomposed BCNF relations:

SERVICES_0(RADDR, SADDR, AMOUNT, PINCODE, CONSIGNMENTNO) having FD(s): CONSIGNMENTNO → RADDR; CONSIGNMENTNO → SADDR; CONSIGNMENTNO → AMOUNT; CONSIGNMENTNO → PINCODE; PINCODE → RADDR; PINCODE → SADDR; PINCODE → PINCODE; PINCODE → CONSIGNMENTNO.

SERVICES_1(ITEM, WEIGHT, CONSIGNMENTNO) having FD(s): (none).

SERVICES_2(RADDR, SADDR, AMOUNT, PINCODE, CONSIGNMENTNO) having FD(s): CONSIGNMENTNO → RADDR; CONSIGNMENTNO → SADDR; CONSIGNMENTNO → AMOUNT; CONSIGNMENTNO → PINCODE; PINCODE → RADDR; PINCODE → SADDR; PINCODE → PINCODE; PINCODE → CONSIGNMENTNO.

SERVICES_3(ITEM, WEIGHT, PINCODE) having FD(s): PINCODE → PINCODE.

Final set of decomposed BCNF relations (removing duplicate and subset relations):

SERVICES_0(RADDR, SADDR, AMOUNT, PINCODE, CONSIGNMENTNO) having FD(s): CONSIGNMENTNO → RADDR; CONSIGNMENTNO → SADDR; CONSIGNMENTNO → AMOUNT; CONSIGNMENTNO → PINCODE; PINCODE →

RADDR; PINCODE → SADDR; PINCODE → PINCODE; PINCODE → CONSIGNMENTNO.

SERVICES_1(ITEM, WEIGHT, CONSIGNMENTNO) having FD(s): (none).

SERVICES_3(ITEM, WEIGHT, PINCODE) having FD(s): PINCODE → PINCODE.

No input functional dependencies were lost.

Decomposing input relation into BCNF relations using decomposed 3NF relations and the minimal cover set of functional dependencies as sources.

Final set of decomposed BCNF relations:

SERVICES0(AMOUNT, PINCODE, CONSIGNMENTNO) having FD(s): CONSIGNMENTNO → AMOUNT; CONSIGNMENTNO → PINCODE; PINCODE → CONSIGNMENTNO.

SERVICES1(RADDR, SADDR, PINCODE, CONSIGNMENTNO) having FD(s): CONSIGNMENTNO → PINCODE; PINCODE → RADDR; PINCODE → SADDR; PINCODE → CONSIGNMENTNO.

SERVICES3(ITEM, WEIGHT, CONSIGNMENTNO) having FD(s): (none).

No functional dependencies from the minimal cover set were lost.

3. Access

access(transactionid,consignmentid,aadhar,date)

transactionid,consignmentno -> aadhar,date
transactionid -> consignmentno

Determining highest normal form of relation:

Input relation is assumed to be in 1NF: each attribute is assumed to contain only one value per row.

Input relation is in 2NF: It is in 1NF and there are no composite minimum keys (minimum keys composed of more than one attribute).

Input relation is in 3NF: It is in 2NF and for each functional dependency: (1) The right-hand side is a subset of the left hand side, (2) the left-hand side is a superkey (or minimum key) of the relation, or (3) the right-hand side is (or is a part of) some minimum key of the relation.

Input relation is in BCNF: it is in 3NF and for each functional dependency: (1) The right-hand side is a subset of the left hand side, or (2) the left-hand side is a superkey (or minimum key) of the relation.

Input relation is in 4NF: it is in BCNF and for each of its nontrivial multivalued dependencies: the left-hand side is a superkey (or minimum key) of the relation. A multivalued dependency is trivial if either (1) the right-hand side is a subset of the left-hand side, or (2) the multivalued dependency contains all attributes of the input relation.

Decomposing input relation into 3NF using canonical functional dependency cover (lossless and preserving all minimal cover set functional dependencies):

Input relation is already in 3NF. No decomposition necessary.

Decomposing input relation into BCNF relations (lossless but not necessarily functional dependency preserving). Will attempt two parallel decompositions: one from the input relation and the second from the set of decomposed 3NF relations:

Input relation is already in BCNF. No decomposition necessary.

4. Customer

customer(aadhar,name,email,address,phone)
aadhar-> name,email,address,phone

Determining highest normal form of relation:

Input relation is assumed to be in 1NF: each attribute is assumed to contain only one value per row.

Input relation is in 2NF: It is in 1NF and there are no composite minimum keys (minimum keys composed of more than one attribute).

Input relation is in 3NF: It is in 2NF and for each functional dependency: (1) The right-hand side is a subset of the left hand side, (2) the left-hand side is a superkey (or minimum key) of the relation, or (3) the right-hand side is (or is a part of) some minimum key of the relation.

Input relation is in BCNF: it is in 3NF and for each functional dependency: (1) The right-hand side is a subset of the left hand side, or (2) the left-hand side is a superkey (or minimum key) of the relation.

Input relation is in 4NF: it is in BCNF and for each of its nontrivial multivalued dependencies: the left-hand side is a superkey (or minimum key) of the relation. A multivalued dependency is trivial if either (1) the right-hand side is a subset of the left-hand side, or (2) the multivalued dependency contains all attributes of the input relation.

Decomposing input relation into 3NF using canonical functional dependency cover (lossless and preserving all minimal cover set functional dependencies):

Input relation is already in 3NF. No decomposition necessary.

Decomposing input relation into BCNF relations (lossless but not necessarily functional dependency preserving). Will attempt two parallel decompositions: one from the input relation and the second from the set of decomposed 3NF relations:

Input relation is already in BCNF. No decomposition necessary.

5. Speedpost

speedpost(consignmentno,raddr,saddr,amount,item,weight)

consignmentno -> raddr,saddr,amount,item,weight

Determining highest normal form of relation:

Input relation is assumed to be in 1NF: each attribute is assumed to contain only one value per row.

Input relation is in 2NF: It is in 1NF and there are no composite minimum keys (minimum keys composed of more than one attribute).

Input relation is in 3NF: It is in 2NF and for each functional dependency: (1) The right-hand side is a subset of the left hand side, (2) the left-hand side is a superkey (or minimum key) of the relation, or (3) the right-hand side is (or is a part of) some minimum key of the relation.

Input relation is in BCNF: it is in 3NF and for each functional dependency: (1) The right-hand side is a subset of the left hand side, or (2) the left-hand side is a superkey (or minimum key) of the relation.

Input relation is in 4NF: it is in BCNF and for each of its nontrivial multivalued dependencies: the left-hand side is a superkey (or minimum key) of the relation. A multivalued dependency is trivial if either (1) the right-hand side is a subset of the left-hand side, or (2) the multivalued dependency contains all attributes of the input relation.

Decomposing input relation into 3NF using canonical functional dependency cover (lossless and preserving all minimal cover set functional dependencies):

Input relation is already in 3NF. No decomposition necessary.

Decomposing input relation into BCNF relations (lossless but not necessarily functional dependency preserving). Will attempt two parallel decompositions: one from the input relation and the second from the set of decomposed 3NF relations:

Input relation is already in BCNF. No decomposition necessary.

6. Courier

consignmentno(raddr,saddr,amount,item,weight,country)
consignmentno -> raddr,saddr,amount,item,weight,country

Determining highest normal form of relation:

Input relation is assumed to be in 1NF: each attribute is assumed to contain only one value per row.

Input relation is in 2NF: It is in 1NF and there are no composite minimum keys (minimum keys composed of more than one attribute).

Input relation is in 3NF: It is in 2NF and for each functional dependency: (1) The right-hand side is a subset of the left hand side, (2) the left-hand side is a superkey (or minimum key) of the relation, or (3) the right-hand side is (or is a part of) some minimum key of the relation.

Input relation is in BCNF: it is in 3NF and for each functional dependency: (1) The right-hand side is a subset of the left hand side, or (2) the left-hand side is a superkey (or minimum key) of the relation.

Input relation is in 4NF: it is in BCNF and for each of its nontrivial multivalued dependencies: the left-hand side is a superkey (or minimum key) of the relation. A multivalued dependency is trivial if either (1) the right-hand side is a subset of the left-hand side, or (2) the multivalued dependency contains all attributes of the input relation.

Decomposing input relation into 3NF using canonical functional dependency cover (lossless and preserving all minimal cover set functional dependencies):

Input relation is already in 3NF. No decomposition necessary.

Decomposing input relation into BCNF relations (lossless but not necessarily functional dependency preserving). Will attempt two parallel decompositions: one from the input relation and the second from the set of decomposed 3NF relations:

Input relation is already in BCNF. No decomposition necessary.

3.vi Table Creation

create table employee(eid number(3) primary key not null, name varchar2(10) not null, designation varchar2(10) not null, salary number(6) not null, address varchar2(12) not null, phone number(10),dob date, pincode number(6) references postoffice(pincode) on delete cascade not null);

Name	Null?	Type
EID	NOT NULL	NUMBER(3)
NAME	NOT NULL	VARCHAR2(10)
DESIGNATION	NOT NULL	VARCHAR2(10)
SALARY	NOT NULL	NUMBER(6)
ADDRESS	NOT NULL	VARCHAR2(12)
PHONE		NUMBER(10)
DOB		DATE
PINCODE	NOT NULL	NUMBER(6)

create table customers(aadhar number(5) primary key not null, name varchar2(10) not null, email varchar2(15) not null, phone number(10), address varchar2(10) not null);

Name	Null?	Type
AADHAR	NOT NULL	NUMBER(5)
NAME	NOT NULL	VARCHAR2(10)
EMAIL	NOT NULL	VARCHAR2(15)
PHONE		NUMBER(10)
ADDRESS	NOT NULL	VARCHAR2(10)

create table postoffice(pincode number(6) primary key not null, address varchar2(12) not null);

Name	Null?	Type
PINCODE	NOT NULL	NUMBER(6)
ADDRESS	NOT NULL	VARCHAR2(12)

create table services(consignment_no number(3) primary key not null, aadhar number(5) references customers(aadhar) on delete cascade not null, pincode number(6) references postoffice(pincode) on delete cascade not null, r_addr varchar2(10) not null, s_addr varchar2(10) not null, amount number(6) not null);

Name Null? Type

CONSIGNMENT_NO NOT NULL NUMBER(3)

PINCODE NOT NULL NUMBER(6)

R_ADDR NOT NULL VARCHAR2(10)

S_ADDR NOT NULL VARCHAR2(10)

AMOUNT NOT NULL NUMBER(6)

create table accesst(transaction_id number(3) primary key not null, aadhar number(5) references customers(aadhar) on delete cascade not null, consignment_no number(3) references services(consignment_no) on delete cascade not null, dateT date not null);

Name Null? Type

TRANSACTION_ID NOT NULL NUMBER(3)

AADHAR NOT NULL NUMBER(5)

CONSIGNMENT_NO NOT NULL NUMBER(3)

DATET NOT NULL DATE

create table speedpost(consignment_no number(3) references services(consignment_no) on delete cascade not null,r_addr varchar2(10) not null, s_addr varchar2(10) not null, amount number(6) not null, item varchar2(10) not null, weight number(3) not null);

Name Null? Type

---------- -------- ----------

CONSIGNMENT_NO NOT NULL NUMBER(3)

R_ADDR NOT NULL VARCHAR2(10)

S_ADDR NOT NULL VARCHAR2(10)

AMOUNT NOT NULL NUMBER(6)

ITEM NOT NULL VARCHAR2(10)

WEIGHT NOT NULL NUMBER(3)

create table courier(consignment_no number(3) references services(consignment_no) on delete cascade not null, r_addr varchar2(10) not null, s_addr varchar2(10) not null, amount number(6) not null, item varchar2(10) not null, weight number(3) not null, country varchar2(12) not null);

Name Null? Type

---------- -------- ----------

CONSIGNMENT_NO NOT NULL NUMBER(3)

R_ADDR NOT NULL VARCHAR2(10)

S_ADDR NOT NULL VARCHAR2(10)

AMOUNT NOT NULL NUMBER(6)

ITEM NOT NULL VARCHAR2(10)

WEIGHT NOT NULL NUMBER(3)

COUNTRY NOT NULL VARCHAR2(12)

```
insert all
into customers values(12345,'Aman','am@gmail.com',8820492229,’Kolkata’)
into customers values(54321,'Ram','rm@gmail.com',6236272220,’Mumbai’)
into customers values(12481,'Bhola','bh@gmail.com',6236274589,’Banglore’)
into customers values(88124,'Usha','us@gmail.com',8812349078,’Delhi’)
into customers values(99123,'Mala','sr@gmail.com',9123093445,’Bhopal’)
select * from dual;

select * from customers;
```

AADHAR	NAME	EMAIL	PHONE	ADDRESS
12345	Aman	am@gmail.com	8820492229	Kolkata
54321	Ram	rm@gmail.com	6236272220	Mumbai
12481	Bhola	bh@gmail.com	6236274589	Banglore
88124	Usha	us@gmail.com	8812349078	Delhi
99123	Mala	sr@gmail.com	9123093445	Bhopal

```
insert all
into postoffice values(700156,'Kolkata')
into postoffice values(700159,'Mumbai')
```

```
into postoffice values(700165,'Delhi')

into postoffice values(715564,'Bhopal')

into postoffice values(801505,'Banglore')

select * from dual;


select * from postoffice;


  PINCODE ADDRESS

---------- ------------

    700156 Kolkata

    700159 Mumbai

    700165 Delhi

    715564 Bhopal

    801505 Banglore


insert all

into services values(100,12345,700156,'Kolkata','Mumbai',300)

into services values(101,54321,700165,'Delhi','Kolkata',544)

into services values(102,12481,715564,'Bhopal','Banglore',412)

into services values(103,88124,700156,'Kolkata','Delhi',300)
```

```
into services values(104,99123,700159,'Mumbai','Banglore',250)

into services values(105,54321,700165,'Delhi','Bhopal',502)

into services values(106,12345,700156,'Kolkata','Bhopal',350)

into services values(107,88124,715564,'Bhopal','Kolkata',250)

select * from dual


select * from services;

CONSIGNMENT_NO     AADHAR    PINCODE R_ADDR     S_ADDR         AMOUNT

-------------- ---------- ---------- ---------- ---------- ----------

         100      12345     700156 Kolkata    Mumbai           300

         101      54321     700165 Delhi      Kolkata          544

         102      12481     715564 Bhopal     Banglore         412

         103      88124     700156 Kolkata    Delhi            300

         104      99123     700159 Mumbai     Banglore         250

         105      54321     700165 Delhi      Bhopal           502

         106      12345     700156 Kolkata    Bhopal           350

         107      88124     715564 Bhopal     Kolkata          250


insert all

into employee values(200,'Aman','Manager',56000,'Kolkata',8820492225,'12-APR-91',700156)
```

into employee values(201,'Ramu','Post man',43000,'Bhopal',8820492265,'12-AUG-80',700159)

into employee values(202,'Sudha','Clerk',22400,'Mumbai',8820492456,'04-MAR-91',700165)

into employee values(203,'Dev','Post man',44000,'Kolkata',8820492459,'04-MAR-92',715564)

into employee values(204,'Shiva','Clerk',25000,'Banglore',8820492450,'14-APR-87',801505)

select * from dual

EID NAME DESIGNATIO SALARY ADDRESS PHONE DOB

---------- ---------- ---------- ---------- ---------- ---------- ---------

PINCODE

---

200 Aman Manager 56000 Kolkata 8820492225 12-APR-19

700156

201 Ramu Post man 43000 Bhopal 8820492265 12-AUG-80

700159

202 Sudha Clerk 22400 Mumbai 8820492456 04-MAR-91

700165

EID NAME DESIGNATIO SALARY ADDRESS PHONE DOB

---------- ---------- ---------- ---------- ---------- ---------- ---------

PINCODE

---

203 Dev Post man 44000 Kolkata 8820492459 04-MAR-92

715564

204 Shiva Clerk 25000 Banglore 8820492450 14-APR-87

801505

```
insert all
into accesst values(140,54321,101,'12-SEP-19')
into accesst values(141,99123,104,'30-DEC-19')
into accesst values(142,12345,106,'01-JAN-19')
into accesst values(143,88124,103,'12-MAR-19')
into accesst values(144,54321,105,'12-FEB-19')
into accesst values(145,88124,107,'12-FEB-19')
select * from dual
```

```
select * from accesst;
```

TRANSACTION_ID AADHAR CONSIGNMENT_NO DATET

```
-------------- ---------- -------------- ---------

        140      54321           101 12-SEP-19

        141      99123           104 30-DEC-19

        142      12345           106 01-JAN-19

        143      88124           103 12-MAR-19

        144      54321           105 12-FEB-19

        145      88124           107 12-FEB-19
```

```
insert all
  2  into speedpost values(101,'Laptop',3,54321)
  3  into speedpost values(104,'fan',4,99123)
  4  into speedpost values(106,'guitar',2,12345)
  5  select * from dual;
```

```
CONSIGNMENT_NO ITEM          WEIGHT     AADHAR
-------------- ---------- ---------- ----------
        101 Laptop             3      54321
        104 fan                4      99123
        106 guitar             2      12345
```

```
insert all
```

into courier values(103,'Fridge',60,'India',88124)

into courier values(105,'AC',25,'India',54321)

into courier values(107,'VIP',5,'India',88124)

select * from dual;

CONSIGNMENT_NO ITEM WEIGHT COUNTRY AADHAR

---------- ---------- ---------- ---------- ----------

103 Fridge 60 India 88124

105 AC 25 India 54321

107 VIP 5 India 88124

3.vii Queries using SQL Plus

1.How many transactions were done by ‘Ram’ in 2019?

Ans:-

select count(*) "No. of transactions" from (select name,aadhar,transaction_id,datet from (select transaction_id,aadhar,name,datet from customers natural join accesst) where name='Ram' and datet like '%19');

No. of transactions

2

2. What is the total salary given to post man by the post office ?

Ans:-

```
select sum(salary) from employee where designation='Post man';
```

```
SUM(SALARY)
-----------
      87000
```

3. What is the total money spent by ‘Ram’ ?

Ans:-

```
select sum(amount) from (select aadhar,name,r_addr,s_addr,amount from customers natural join
services) where name='Ram';
```

```
SUM(AMOUNT)
-----------
```

1046

4. What is the total revenue of post office with pincode 700156 and 700159 ?

Ans:-

```
select sum(amount) "Total Revenue" from services where pincode in(700156,700159);
```

```
Total Revenue
-------------
        1200
```

5.What is the the amount paid on different transactions and on which date ?

Ans:-

```
select transaction_id, datet,amount from services natural join accesst;
```

```
TRANSACTION_ID DATET         AMOUNT
-------------- --------- ----------
           140 12-SEP-19        544
           143 12-MAR-19        300
           141 30-DEC-19        250
```

144 12-FEB-19 502

142 01-JAN-19 350

145 12-FEB-19 250

6. List the number of services provided by each post office.

Ans:-

select * from postoffice natural join (select pincode,count(*) "No of services provided" from services group by pincode);

PINCODE	ADDRESS	No of services provided
715564	Bhopal	2
700156	Kolkata	3
700159	Mumbai	1
700165	Delhi	2

4. Conclusion

The post office database that this Requirements Analysis is the process of defining the expectations of the users for an application that is to be built or modified. Requirements analysis involves all the tasks that are conducted to identify the needs of different stakeholders. Therefore requirements analysis means to analyze, document, validate and manage software or system requirements. High-quality requirements are documented, actionable, measurable, testable, traceable, helps to identify business opportunities, and are defined to a facilitate system design.

5. References

- https://www.indiapost.gov.in/MBE/Pages/Content/More-Info.aspx
- *Database System Concepts* by Sudarshan, Korth
- https://docs.oracle.com/en/
- https://www.oracle.com/technetwork/indexes/documentation/index-100966.html
- https://www.w3schools.com/sql/