JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, Noida

Department of CSE & IT

Project Report

Database Systems and Web

(15B17CI372)

Tarang – Serving Humanity

ODD SEM 2018

Submitted By:

Utsav Gupta – 17103040

Rohan Jain – 17103056

Rajat Kumar Garg – 17103062

Hardik Bhardwaj - 17103064

(Batch – B10)

Submitted To:

Mrs. Megha Rathi

(CSE Faculty)

INTRODUCTION

Tarang, an NGO with its two Sub-bodies named **Kutumb** and **Parindey**.

Parindey works for the Literacy of Slums' children.

It is Parindey's motto to impart basic education to under privileged children. The emphasis is on providing education not in a typical classroom manner, but to give a child a wider canvas of exposure in a socio-cultural environment. Literacy India envisions to keep its student's at par with the fast evolving world around them.

In line with this thought, Literacy India is looking at transforming the education landscape through exposing the student to a more innovative and non-traditional method of learning

Eminent personalities from different walks of life come and interact with the children and prove to be a catalyst to the child's motivation and enthusiasm to do well. The interest generated in these activities also helps identify talent which, with further guidance can pursue further education or vocational training.

Kutumb works for the Old Age People providing them with Old Age Homes. All human beings are created in the image of God, and are of equal worth. Therefore the poor have the same inherent dignity, and rights that deserve our protection.

Kutumb is run by a team of dedicated social workers. The organization is devoted to the service of needy Indian citizens aged over 60. Our attempt is to provide every facility and opportunity to the elderly to enable them to live with dignity and hope in a homely environment; where they can feel secure, comfortable and contented. This old age home provides them with shelter, food, medical treatment and entertainment. Doctors are available every day to provide allopathic and homeopathic treatments.

Kutumb continues the saga of growth. At present there are 80 inmates, who are provided with food, accommodation, entertainment and medical facilities. Our endeavor is to provide every facility to the elderly to enable them to live a comfortable and dignified life in a homely environment. The trustee and volunteers are committed and caring individuals. They provide their services free. Most of the employees stat with the organization because they are proud to serve it. The public has also donated generously and has made available many facilities for the welfare of the aged. We extend our gratitude to all for their help. Today, let us all resolve to serve the poorest of the poor, aged and the needy.

Normalisation of Data

Initially, we had a Relation R

R { HqId, Estd, Street, City, State, hqEmail, Contact1, Contact2, CampId, campLocation, MemId, memName, memEmail, memContact, dateJoinMember, gender, dob, bloodgroup, password, childId, childName, guardianName, guardianContact, dateJoinChild, childGender, childDob, childBloodgroup, donationId, Amount, modeOfPayment, donorId, donorName, donorPassword, donorEmail, donorContact, donorGender, oldAgeHomeId, oldAgeLocation, OldAgeMemberId, oldAgeMemberName, oldAgeMemberEmail, oldAgeMemberContact, dateJoinOldAgeMember, OldAgeMember, dobOldAgegender, oldAgeBg, oldAgePassword, oldPeopleId, oldPeopleName, oldPeopleContact, oldPeopledateJoin, oldPeopleGender, oldPeopleDob, oldPeopleBg, reviewerEmail, reviewerName, reviewDate, reviewComment }

Set of Function Dependencies

HqId → Estd, Street, City, State, Email, Contact1, Contact2

CampId → Location, City, HqId, Estd, Street, City, State, Email, Contact1,Contact2

MemId → memName, memEmail, memContact, dateJoinMember, memGender, memDob, memBloodGroup, memPassword, Location, City, HqId, Estd, Street, City, State, Email, Contact1, Contact2

ChildId → childName, guardianName, guardianContact, childDateJoin, childGender, childDob , childBloodGroup, Location, City, HqId Estd, Street, City, State, Email, Contact1, Contact2

DonationId → Amount, modeOfPayment, donorId, donorName, donorPassword, donorEmail, donorContact,donorGender

DonorId → donorName, donorPassword, donorEmail, donorContact,donorGender

OldAgeHomeId → oldAgeHomeLocation, oldAgeHomeCity, Estd, Street, City, State, Email, Contact1, Contact2

OldAgeMemId → oldAgeMemId, oldAgeMemEmail, oldAgeMemContact, oldAgeMemDateJoin, oldAgeHomeMemGender, oldAgeHomeDob, oldAgeHomeBloodGroup,oldAgeMemPassword, oldAgeHomeLocation, oldAgeHomeCity, Estd, Street, City, State, Email, Contact1, Contact2

OldPeopleId → oldAgeHomeId, oldPeopleName, oldPeopleContact, oldPeopleDateJoin, oldPeopleGender,oldPeopleDob, oldPeopleBloodGroup, oldAgeHomeLocation, oldAgeHomeCity, Estd, Street, City, State, Email, Contact1, Contact2

ReviewEmail → reviewerName, reviewDate, Comment

I Normal Form (Removal of Multi-Valued Attributes)

R1 { HqId, Estd, Street, City, State, Email, CampId, campLocation, MemId, memName, memEmail, memContact, dateJoinMember, gender, dob, bloodgroup, password, childId, childName, guardianName, guardianContact, dateJoinChild, childGender, childDob, childBloodgroup, donationId, Amount, modeOfPayment, donorId, donorName, donorPassword, donorEmail, donorContact, donorGender, oldAgeHomeId, oldAgeLocation, OldAgeMemberId, oldAgeMemberName, oldAgeMemberEmail, oldAgeMemberContact, dateJoinOldAgeMember, OldAgeMember, dobOldAgegender, oldAgeBg, oldAgePassword, oldPeopleId, oldPeopleName, oldPeopleContact, oldPeopledateJoin, oldPeopleGender, oldPeopleDob, oldPeopleBg, reviewerEmail, reviewerName, reviewDate, reviewComment }

R2 { HqId, Contact }

II Normal Form (Removal of Partial Dependencies)

R11 { HqId, Estd, Street, City, State, Email }

R12 { MemId, memName, memEmail, memContact, dateJoinMember, memGender, memDob, memBloodGroup, memPassword, Location, City, HqId, Estd, Street, City, State, Email, Contact1, Contact2 }

R13 { childId, childName, guardianName, guardianContact, childDateJoin, childGender, childDob , childBloodGroup, Location, City, HqId Estd, Street, City, State, Email, Contact1, Contact2 }

R14 { DonationId, Amount, modeOfPayment, donorId, donorName, donorPassword, donorEmail, donorContact, donorGender }

R15 { OldAgeMemId, oldAgeMemId, oldAgeMemEmail, oldAgeMemContact, oldAgeHomeMemDob, oldAgeMemDateJoin, oldAgeHomeMemGender, oldAgeHomeMemBloodGroup, oldAgeMemPassword, oldAgeHomeCity }

R16 { OldPeopleId, oldAgeHomeId, oldPeopleName, oldPeopleContact, oldPeopleDateJoin, oldPeopleGender,oldPeopleDob, oldPeopleBloodGroup, oldAgeHomeLocation, oldAgeHomeCity }

R17 { ReviewEmail, reviewerName, reviewDate, Comment }

R2 { HqId, Contact }

<u>III Normal Form (Removal of Transitivity)</u>

R1 **Headquarters** { <u>HqId</u>, Estd, Street, City, State, Email }

R2 **HeadquarterContact** {<u>HqId</u>, Contact }

R3 **CampDetails** {<u> CampId</u>, HqId, Location, City }

R4 **CampMembers** { <u>MemId</u>, CampId, MemName, MemEmailId, MemContact, MemDateJoin, MemGender, MemDob, MemBloodGroup, MemPassword }

R5 **Children** { <u>childId</u>, childName, guardianName, guardianContact, childDateJoin, childGender, childDob , childBloodGroup, CampId}

R6 **Donations** { <u>donationId</u>, HqId, donorId, , Amount, ModeofPayment }

R7 **Donors** { <u>donorId</u>, donorEmail, donorContact, donorGender }

R8 **Helps** { <u>Memid, Oldpeopleid</u>}

R9 **OldAgeHomes** {<u>HomeId</u>, Hqid, location, city}

R10 **OldAgeMembers** {Memid, OldageHomeid, MemName, MemEmailId, MemContact,
MemDateJoin, MemGender, MemDob, MemBloodGroup,
MemPassword }

R11 **OldPeople** {OldPeopleId, OldageHomeid, OldPeopleName,
OldPeopleContact,OldPeopleDateJoin, OldPeopleGender,
OldPeopleDob, OldPeopleBloodGroup}

R12 **Review** { reviewerEmail, reviewerName, reviewDate, reviewComment }

R13 **Teaches** { MemId, ChildId }

R2 **HqContacts** { HqId, Contact }

PL/SQL

PROCEDURES :-

(1) Procedure to display the details of camp members in a given city.

```
Delimiter $
create procedure CampMembersInCity(in cityin varchar(20))
begin
select memid,name,email,contact from campmembers where camped in (select
id from campdetails where city=cityin);
end$
```

(2) Procedure to display the details of Old Age Home members in a given city.

```
Delimiter $
create procedure HomeMembersInCity(in cityin varchar(20))
begin
select memid,name,email,contact from oldagemembers where oldagehomeid in
(select id from oldagehomes where city=cityin);
end$
```

(3) Procedure to display the details of Camps in a given State.

```
Delimiter $
create procedure CampsInState(in statein varchar(20))
begin
select id,location,city from campedetails where hqid in (select hqid from
headquarters where state=statein);
end$
```

(4) Procedure to display the details of Old Age Homes in a given State.

```
Delimiter $
create procedure HomesInState(in statein varchar(20))
begin
select id,location,city from oldagehomes where hqid in (select hqid from
headquarters where state=statein);
end$
```

PL/SQL TRIGGERS AND CURSORS:-

Trigger to display error message when a primary key of a table is trying to be updated.

```
Delimiter $
create trigger error1
before update on campmembers
for each row
begin
If new.memid<>old.memid then
signal sqlstate '45000' set message_text="Can’t update User ID";
end if;
end$

create trigger error2
before update on oldagemembers
for each row
begin
If new.memid<>old.memid then
signal sqlstate '45000' set message_text="Can’t update User ID";
end if;
end$

create trigger error3
before update on donors
for each row
begin
If new.id<>old.id then
signal sqlstate '45000' set message_text="Can’t update Donor ID";
end if;
end$
```

Cursor to assign a member to a child who has no mentor

```
Create procedure assignmem()
Begin
Declare done int default 0;
Declare childid int;
Declare memid int;
Declare campid int;
Declare cur1 cursor for select id,campid from children where id not in select childid from Helps;
Declare continue handler for not found set done=1;
Open cur1;
Repeat
Fetch cur1 into childid,campid;
Select memid into memid from campmembers inner join children on Campmembers.campid=children.campid limit 1;
Insert into teaches values(memid,childid);
Until done;
End repeat;
Close cur1;
End
$
```

Learning Outcomes

1. Better Understanding and practical Implementations of Javascript and PHP
2. Better Understanding the concept of Normalisation
3. Conversion of Problems into ER Models and Equivalent Relational Model
4. Better Understanding of Database Connectivity with Web Pages
5. Front End Development using HTML and CSS