Project Report

Immigration Management System

Group Members:

Dhairya Chauhan	18BCP024
Falak Panchal	18BCP031
Heet Sinojiya	18BCP035
Jay Patel	18BCP043
Jay Hemnani	18BCP045
Dhruvil Soni	18BCP143D

Index

1. Advantages of DBMS over file system:

1) Data redundancy and inconsistency:

Data redundancy mainly as the name suggests, there won't be any data's duplication in it such that those duplicate values must differ from each other. If the programmer make the files or the application, they are in different file formats. Also some part of information could be duplicated in some files.
For instance, the address and phone number of a particular customer may appear in two tables and due to this higher storage is required and access cost.

2) Difficulty in accessing data:

If the head of the visa consultancy needs to find the total number of people who came there in some particular month then file system isn't advised. As the programmer has to develop or edit the program and it would take some time or he can take the list of all customers and can check it manually. So these two options can be done. After some time if he says that he wants to check customer who was for visitor visa then again he has those two paths and none of them is advisable.

3) Data isolation:

As the data stored are in different file formats and to get the proper data is difficult if we write new program everytime.
In DBMS one file would be there and in the same format such that the application program can give the required output.

4) Integrity problems:

It mainly states that the data in our DBMS must deal with a minimum requirement or some kind of rules.
For example, the customer visiting for the visitor visa should have a minimum (some required) bank-balance and properties.

5) Atomicity problem:

The concept all-or-none requirement is called atomicity.
Here if the fees from the customer end are submitted to agency then both should be satisfied. That is amount must be received by the agency and must be debited from customer.

6) Concurrent access anomalies:

In order to statisfy the person's needs i.e. fast performance of the system and quicker output, multiple users uses the system to update the information.
There can be two employees who deal with the edition work and if employee A did some edition then employee B must see that and It shouldn't be like, only the edition done by employee A is shown or by employee B is shown.

7) Security problems:

There should'nt be all permission to every user to get access of most of the data. And in the file system i.e., in applications it is arduous to maintain security problems.
If the personal information or the customer is only for the head of the agency and not for the employees then in file system there is nothing like that both can see the personal information, and so there is lack of security. This could be overcome by DBMS.

Hence, these difficulties can be overcome only by the help of DBMS.

Tables though before the ER and relational model:

1.inquiry
inq_id,date_of_inq ,interested,person_id

2.reference
reference_id,discount,reference_type,person_id

3.registration
reg_id,date_of_reg,inq_id

4. type_of_visa
visa_type,reg_id,visatype_id

5.student_visa
country,age,stud_id

6.business_visa
work,cs_id,b_duration,bus_id

7.visitor_visa
travel_agency,cs_id,v_duration,visit_id

8.course
course_id,course_type,stud_id,

9.country_state
cs_id, country, state

10. result
result_id, ielts_score, hsc_percent, ug_cgpa, bba_cgpa, university

11. business_documents

b_passport_no,application_form,application_fee_receipt,tax_statments,ps_photo ,identity_proof,bus_id

12.visitor_documents

v_passport_no,application_form,application_fee_receipt,tax_statements,ps_phot o,identity_proof,visit_id

13.student_documents

identity_proof,s_passport_no,acceptance_proof,application_fee_receipt,ps_photo,financial_support,stud_id

14.feedback
feedback_id,overall,service,nature_of_emp,efforts_of_emp,reg_id

15.payment
payment_id, payment_type,reg_id

16.character
character_id,comitted_crime,criminal_record_submitted,reg_id

17. b_confirmation
b_confirmed, bus_id

18. s_confirmation
s_confirmed, stud_id

19. v_confirmation
v_confirmed, visit_id

20. personal_information
person_id,email_id, dob, first_name, middle_name, last_name, gender, phone_no, marital_status,address, state

2. Relational Algebra

Select Operation:

The select operation is able to select the tupleswhich are given to be part of the output. It is denoted by symbol **σ.** And the query tuples to be found are written in subscript of it.

Project Operation:

If the user want to select the some kind of specific attributes from the table inspite of all the attributes from table then project operation is used. It is denoted by symbol **π.**

Cartesian Product Operation:

It is denoted by a (**X**) and it allows us to deal with the combination of information from any two relations.

Union:

A union denoted by **U** symbol. It consists all the tuples that are part of table 1 and table 2. It also removes duplicate values.

Set-Difference:

(-) symbol denotes it. The answer of J **-** K, is a relation that have all tuples that are in J but not in K.

Natural Join:

We can use natural join only when there is a common column between two tables. Also the name and attribute's type should be the same.
The symbol for it is **⋈.**

Select all the data from a business visa table where country name is USA

$\sigma_{country=”usa”}(business\_visa)$

Select course_type- from course table.

$\pi_{course\_type}(course)$

Select stud_id from student_visa and course table for same values of it.

$\sigma_{student\_visa.stud\_id=course.stud\_id}(student\_visa \times course)$

Display all the reg_id from student_visa table and business_visa.

$\pi_{reg\_id}(country(student\_visa))$ $\pi_{reg\_id}(country(business\_visa))$

Get unique reg_id from student_visa and business_visa.

$\pi_{reg\_id}(country(student\_visa)) - \pi_{reg\_id}(country(business\_id))$

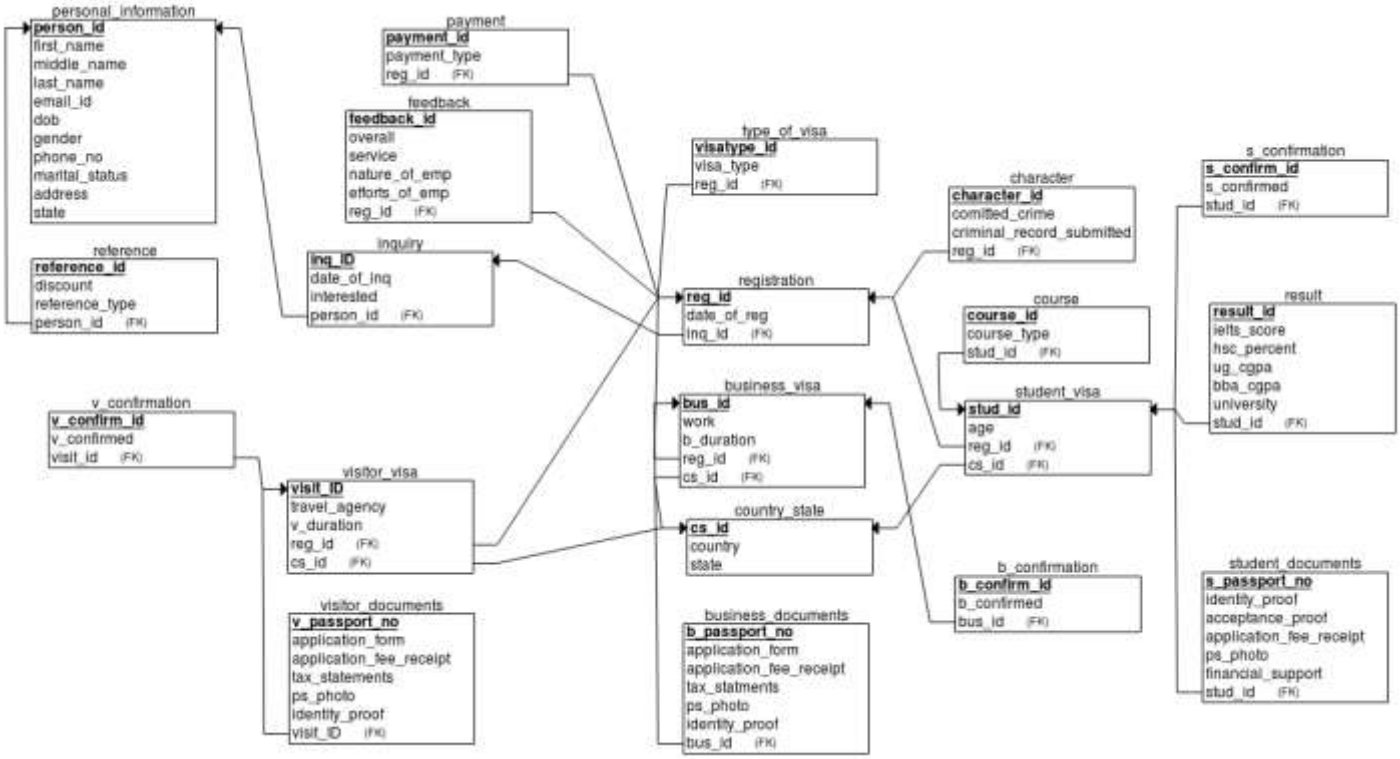
Display age, reg_id, course_id, stud_id, country and course_type from the student_visa and course table.

$student\_visa \bowtie course$

3. Relational Model:

Relations that exists:

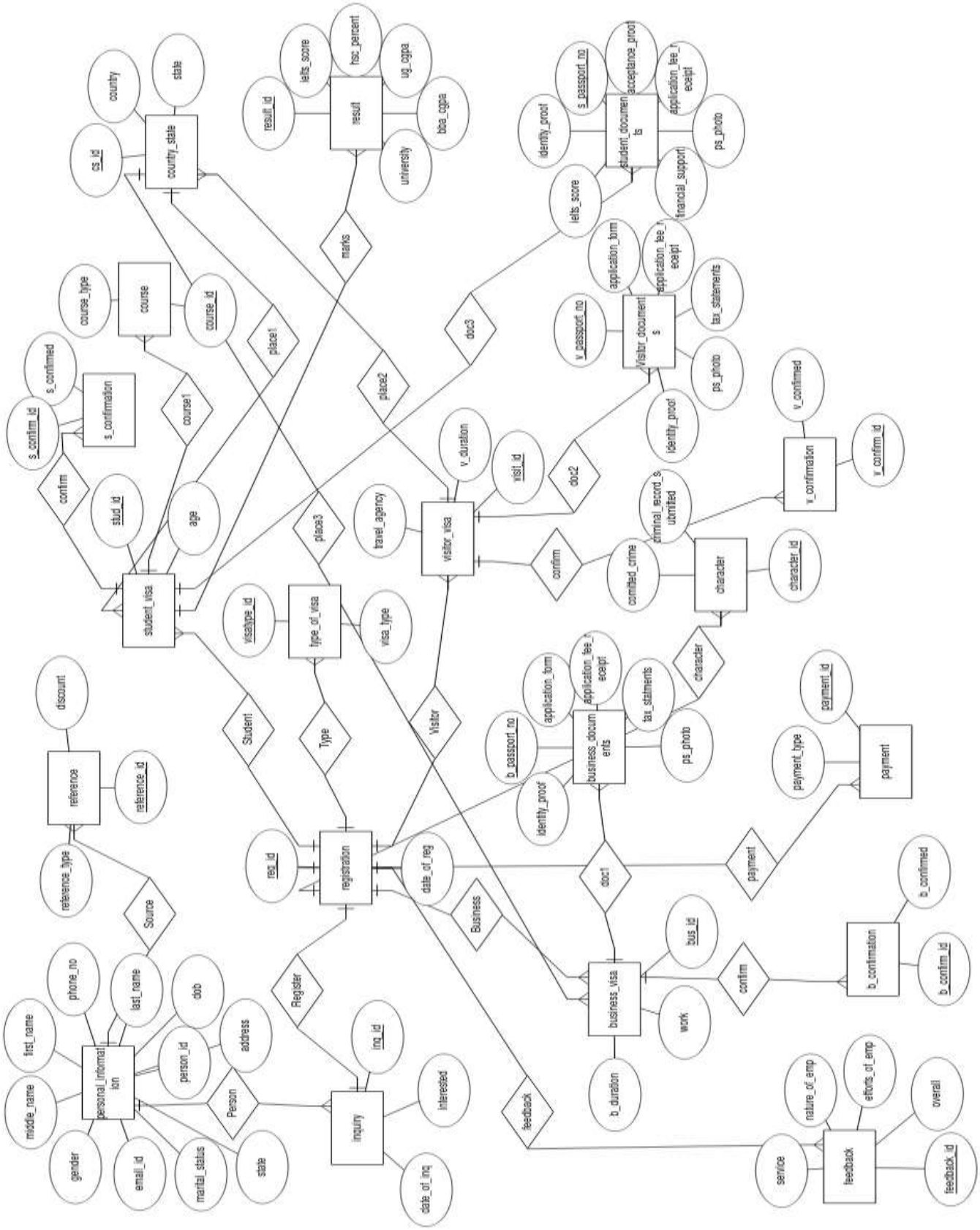
- Personal Information is related by his/her reference, Inquiry date.
- If interested then he/she is registered.
- Registration is related to the type of visa, student visa, business visa, visitor visa, feedback, payment, and character.
- Payment is related to confirmation of visa.
- Visa type is related to Documents for the confirmation of visa.
- A student visa is related to courses like Btech, MBA, Mtech.



.

Here, FK= Foreign Key

4. Entity Relational Model:



Primary Keys:

- person_id: In personal information, it is only unique every time and can identify the person uniquely.
- inq_id: When a person comes for the purpose of inquiry at that time it is generated and is unique for all people.
- reg_id: If the person is interested to confirm the visa and has paid the fees then registration ID will be generated for that person.
- stud_id: When a person visits for a student visa then a unique student ID is generated for him/her.
- bus_id: It is the business ID generated for the person who is willing to visiting other countries for business purposes.
- visit_id: When someone wants to visit the foreign country for some duration then visit_id is using to uniquely identify him/her from DB.
- v_passport_no: When he/she submits the documents of visitor visa then passport_no is also to be submitted.
- s_passport_no: When he/she submits the documents of student visa then passport_no is also to be submitted.
- b_passport_no: When he/she submits the documents of business visa then passport_no is also to be submitted.
- reference_id: From where the customer got the reference to this service.
- v_confirm_id: Whenever the visitor visa is confirmed it would be autogenerated.
- payment_id: When the payment is done by the customer it is an only unique way to identify that person.
- feedback_id: How the service was to the customer has to be submitted by them and feedback_id is generated.
- visatype_id: Which type of visa the customer is interested to go with.
- course_id: The type, of course, that is for student visa.
- character_id: The documents of a criminal report if there.
- cs_id: The country and state of the customer would be identified by it.

Foreign Keys:

- person_id: It is used as a foreign key in the reference table and person table.
- inq_id: It is used in person table, registration table.
- reg_id: In visitor, business, student, payment, character tables.
- stud_id: In Btech, Mtech, MBA, course, document tables, and confirmation table.
- bus_id: In business document table and confirmation table.
- visit_id: In visitor visa document table and confirmation table.
- cs_id: It is mainly for the visitor visa, student visa, and business visa tables.

5. Create Table in SQL

There is code for all the 20 tables but few of them is listed below:

```
create table personal_information
(
person_id int(10) NOT NULL,
first_name char(20) NOT NULL,
middle_name char(20) NOT NULL,
last_name char(20) NOT NULL,
email_id varchar(50) NOT NULL,
dob date NOT NULL,
gender char(4) NOT NULL,
mobile bigint(10) NOT NULL,
marital_status char(4) NOT NULL,
primary key (person_id)
);


create table course
(
course_id int(10) NOT NULL,
course_type varchar(20) NOT NULL,
stud_id int(10) NOT NULL,
primary key (course_id),
foreign key (stud_id) references student_visa(stud_id)
);


create table country_state
(
cs_id int(10) NOT NULL,
country varchar(20) NOT NULL,
state varchar(20) NOT NULL,
primary key (cs_id)

);


create table result
(
```

```
result_id int(10) NOT NULL,
university varchar(20) NOT NULL,
ielts_score double(2,2) NOT NULL,
hsc_percent varchar(3),
ug_cgpa double(2,2),
bba_cgpa double(2,2) ,
primary key (result_id),
foreign key (stud_id) references student_visa(stud_id)
);
```

6. Functional Dependencies and their Normal forms

personal_information:

person_id → p1
first_name → p2
middle_name → p3
last_name → p4
email_id → p5
dob → p6
gender → p7
phone_no → p8
marital_status → p9
Address→ p10
State → p11

Functional dependencies

p1 → p2 p3 p4 p1 → p8 p5 p6 p9 p10 p11 p7 p11 → p10	Union ------>	p1 → p2 p3 p4 p8 p5 p6 p9 p10 p11 p7 p11→ p10

Candidate Key= p1

2NF→ Yes
3NF → Yes
BCNF → Yes

reference:

reference_id → r1
discount → r2
reference_type → r3
person_id → r4

FD

r3 → r2 r1→ r4 r1→ r2 r3	Union ------>	r1→ r4 r2 r3 r3 → r2	Extraneous ---------->	r1→ r4 r3

Candidate Key= r1

2NF→ Yes
3NF → Yes
BCNF → Yes

payment:

payment_id → pa1
payment_type → pa2
reg_id → pa3

FD
pa1 →pa3 union
pa1 →pa2 ------> pa1→ pa2 pa3

Candidate Key=pa1

2NF → Yes
3NF → Yes
BCNF → Yes

feedback:

feedback_id → f1
Overall → f2
Service → f3
nature_of_emp → f4
efforts_of_emp → f5
reg_id → f6

FD

f1 →f6 union f1 →f6 f2
f3 f4 f5 → f2 ------> f3 f4 f5 → f2
f1 →f2

Candidate Key= f1 f3 f4 f5

2NF → No
f(f1,f2,f6) f1 → f6 f2 CK=f1
f(f3,f4,f5,f2) f3 f4 f5 → f2 CK= f3 f4 f5
Now it is 3NF and BCNF also.

student_visa

stud_id → s1
age → s2
reg_id → s3
cs_id → s4

FD

s1→ s3 s4 union s1→ s2 s3 s4 Candidate Key = s1
s1→ s2 ------>

2NF → Yes
3NF → Yes
BCNF → Yes

course

course_id → c1
course_type → c2
stud_id →c3

FD

c1 → c3
c1 → c2 union c1 → c2 c3 Candidate Key = c1
------>

2NF → Yes
3NF → Yes
BCNF → Yes

s_confirmation

s_confirm_id → a
s_confirmed → b
stud_id → c

FD

a → b
a → c union a →bc Candidate Key = a
------>

2NF→ Yes
3NF → Yes
BCNF → Yes

result

result_id →a
ielts_score →b
hsc_percent →c
ug_cgpa →d
bba_cgpa →e
university →f
stud_id →g

FD

B → f
C →f Candidate Key abcde
d →f
e → f
a → g

2NF → No

f(b,f)	b→ f	CK=b
f(c,f)	c→ f	CK=c
f(d,f)	d→ f	CK=d
f(e,f)	E →f	CK=e
f(a,g)	a→ g	CK=a

Now it is 3NF and BCNF also.

student documents

s_passport_no →a
identity_proof → b
acceptance_proof →c
application_fee_recipt →d
ps_photo → e
financial_support →f
stud_id → g

FD

a → b
a → g
a → e
a → f
a →c
a → d

union
---------->

A → bcdefg

Candidate Key a

2NF → Yes
3NF → Yes
BCNF → Yes

v_confirmation

v_confirm_id → a
v_confirmed → b
visit_id → c

FD

a →b
a → c

union
---------->

a →bc

Candidate Key = a

2NF → Yes
3NF → Yes
BCNF → Yes

visitor_documents

v_passport_no → a
application_form → b
application_fee_recipt → c
tax_statements → d
ps_photo → e
identity_proof →f
visit_id → g

FD

a →b
a → g
a → e
a → f
a → c
a → d

union a → bcdefg
--------->

Candidate Key = a

2NF → Yes
3NF → Yes
BCNF → Yes

visitor_visa

visit_id → a
travel_agency → b
reg_id → c
cs_id → d

FD

A → b
a → cd union a → bcd Candidate Key = a
------>
2NF → Yes
3NF → Yes
BCNF → Yes

business_visa

bus_id→a
work→b
b_duration→c
reg_id→d
cs_id→e

FD

a → c
a → d union a → bcde Candidate Key = a
a → b ------>

2NF → Yes
3NF → Yes
BCNF → Yes

Country_state:
cs_id → a
country → b
state → c

FD

a → b Union
---------> a → bc Candidate key- a
a → c

2NF → Yes
3NF → Yes
BCNF → Yes

character:
char_id → a
commited_crime → b
criminal_record_submited → c
reg_id → d

Functional Dependencies:
a → b Union a → bd minimal cover a → bd
---------> b →c ----------> b →c Candidate Key- ab
b → c
a → d

2NF →No
a→bd
and b→c
3NF→yes
BCNF→ Yes

buisness_documents:

b_passport_no → a
application_form → b
application_fee_receipt → c
tax_statements → d
ps_photo → e
identity_proof → f
bus_id → g

FD

a → b union a → bg minimal cover a → bg
b →cdef ----------> b cdef ----------> b → cdef Candidate Key- ab
a → g

2NF → No
a →bg and b → cdef
3NF → Yes
BCNF → Yes

inquiry:
inq_id → a
date_of_inq → b
interested → c
person_id → d

Functional Dependencies:
a → b union
a → c --------> a → bcd Candidate key- a
a → d

2NF → Yes
3NF → Yes
BCNF → Yes

type_of_visa:
visatype_id → a
visa_type → b
reg_id → c

FD:

a → b
union a → bc Candidate Key = a
a → c ------>

2NF → Yes
3NF → Yes
BCNF → Yes

registration:
reg_id → a
rate_of_reg → b
inq_id → c

FD:

a →b union a →bc Candidate Key = a

a → c ------>

2NF → Yes

3NF → Yes

BCNF → Yes

b_confirmation:

b_confirm_id → a

b_confirmed → b

Bus_id → c

a→ b union a → bc candidate Key = a

a → c ------>

2NF → Yes

3NF → Yes

BCNF → Yes

7. Triggers

Basically trigger is a group or set of SQL as well as statements of PL/SQL. It is a procedure that can be called automatically when some kind of updation or deletion is required to go with. So it can be helped when we need to have the old mobile number of customer when they change their mobile number. It can be used to prevent misuse of database and to implement business rule constraints, such as balance should not be negative in value.

Syntax for the trigger:

create trigger [name_of_trigger]
[before | after]
{insert | update | delete}
on [name_of_table]
[for each row]
[body_of_trigger]

Here,

create trigger [name_of_trigger]: It will create and would replace the trigger with the name_of_trigger.

[before | after]: It will state when to start the trigger.
{insert | update | delete}: It will deal with the data manipulation langugage operation.

on [name_of_table]: Type the table's name on which you need to fire the trigger.

[for each row]: By this trigger will be executed for every rows in the table so that if can update all the data.

[body_of_trigger]: Here we can add the operation we need to do with the trigger.

Now examples related to our project,

```
DROP TRIGGER IF EXISTS `delete_from_personal`;
CREATE DEFINER=`root`@`localhost` TRIGGER `delete_from_personal`
BEFORE DELETE ON `personal_information`
FOR EACH ROW
INSERT INTO deleteinform
SET email=OLD.email,
first_name=OLD.first_name,
last_name=OLD.last_name
```

```
DROP TRIGGER IF EXISTS `update_inform`;
CREATE DEFINER=`root`@`localhost` TRIGGER `update_inform`
BEFORE UPDATE ON `personal_information`
FOR EACH ROW
INSERT INTO updateinform
SET Mobile_No="UPDATE",
new_mobile= NEW.mobile,
old_mobile=OLD.mobile,
email=OLD.email
```

8. Uniqueness of the Project:

- In our project, we can't lose any kind of information about the customer as we can also have their old and new mobile number.
- Here the customer won't face some issues regarding the submission of the documents and we fully guide the customer that which country would be best for them if they are going for the student visa.
- Also, the character of the visitor is fully verified that he/she shouldn't have any criminal record.
- And the best thing is security which means personal information of the customers is secured and couldn't be lost.

9. Conclusion:

So this project could be used by most of the agencies rather than using excel because there could be duplication of data as well as more storage would be required. Even if we build the front end for this project then it would of great use to them as some of the frequently asked queries could be solved directly.

10. References:

- www.guru99.com
- www.geeksfotgeeks.org