**CSC540 Project 1 Final Report**

**November 3, 2015**

1. **Team Members:**

* Chidambara Bharathi Subbiah Velu Angamuthu (csubbia)
* Nivasse Ajagane (najagan)
* Ravishankar Chandhiramoorthi (rchandh)
* Sekharan Muthusamy Natarajan (smnatara)
* Dixon Crews (dccrews)

1. **ER Diagram**

Please refer to the image present in the submitted zip file. The image was too large to fit in one page of the word document. The image file (ER\_COMPLETED) is stored in the zip folder.

1. **Relational Model**
2. The list of tables and their description (Note: FD stands for Functional Dependency):
3. **Table Name: Nationality**

Description: Stores the list of the countries from which the students have come.

Constraints: Primary key constraint is enforced on the country\_name column (which automatically enforces the NOT NULL and UNIQUE constraints)

FD: country\_name->country\_name

SQL:

**CREATE** **TABLE** Nationality (

country\_name varchar2(50) **NOT** **NULL**,

**CONSTRAINT** pk\_country\_name **PRIMARY** **KEY** (country\_name)

);

1. **Table Name: Degree\_Program**

Description: Stores the complete list of degree programs offered at NCState University.

Constraints: Primary key constraint is enforced on the program column

FD: program->program

SQL:

**CREATE** **TABLE** Degree\_Program (

program varchar2(25) **NOT** **NULL**,

**CONSTRAINT** pk\_degree\_program **PRIMARY** **KEY** (program)

);

1. **Table Name: Faculty\_Category**

Description: Stores the list of various faculty categories that are present at NCState.

Constraints: Primary key constraint is enforced on the category column

FD: category->category

SQL:

**CREATE** **TABLE** Faculty\_Category (

category varchar2(25) **NOT** **NULL**,

**CONSTRAINT** pk\_faculty\_category **PRIMARY** **KEY** (category)

);

1. **Table Name: Departments**

Description: Stores the list of departments (with a unique abbreviation and the name of the department) that are present at NCState.

Constraints: Primary key constraint is enforced on the abbreviation column

FD: abbreviation -> abbreviation, name

SQL:

**CREATE** **TABLE** Departments (

abbreviation varchar2(5),

name varchar2(50),

**CONSTRAINT** pk\_departments **PRIMARY** **KEY** (abbreviation)

);

1. **Table Name: Courses**

Description: Stores the list of courses (with a unique abbreviation and the name of the department) that are being offered at NCState.

Constraints: Primary key constraint is enforced on the (id, dep\_abbreviation) columns.

Foreign key constraint is enforced on the dep\_abbreviation column which references the dep\_abbreviation column in Departments table

FD: id, dep\_abbreviation -> id, name, dep\_abbreviation

**SQL:**

**CREATE** **TABLE** Courses (

id number(10),

name varchar2(100),

dep\_abbreviation varchar2(5),

**CONSTRAINT** pk\_courses **PRIMARY** **KEY** (id, dep\_abbreviation),

**CONSTRAINT** fk\_courses\_departments **FOREIGN** **KEY** (dep\_abbreviation) **REFERENCES** Departments (abbreviation)

**ON** **DELETE** **CASCADE**);

1. **Table Name: Patron**

Description: Stores the names and basic details of every student/faculty at NCSU.

Constraints: Primary key constraint is enforced on the id column.

Foreign key constraint is enforced on the country\_name column which references the country\_name column in Nationality

The CHECK constraint is enforced on the status column to check if it is only holding acceptable string values: ‘GOOD’ or ‘BAD’

FD: id -> fname, lname, id, status, country\_name

SQL:

**CREATE** **TABLE** Patron(

fname varchar2(25) **NOT** **NULL**,

lname varchar2(25) **NOT** **NULL**,

id number(10) **NOT** **NULL**,

status varchar2(25) **NOT** **NULL**,

country\_name varchar2(50) **NOT** **NULL**,

**CONSTRAINT** fk\_patron\_nationality **FOREIGN** **KEY** (country\_name) **REFERENCES** Nationality (country\_name),

**CONSTRAINT** pk\_patron **PRIMARY** **KEY** (id),

**CONSTRAINT** chk\_status **CHECK**(status **IN** ('GOOD','BAD'))

);

1. **Table Name: Student**

Description: Stores the names and basic details of every student at NCSU.

Constraints: Primary key constraint is enforced on the id column.

Foreign key constraint is enforced on the following columns:

* id column which references the id column in Patron table
* program column which references the program column in Degree\_Program table
* dept column which references the abbreviation column in the Departments table

FD: id -> phone, alt\_phone, dob, sex, street, city, postcode, id, program, year, dept

SQL:

**CREATE** **TABLE** Student (

phone varchar2(25) **NOT** **NULL**,

alt\_phone varchar2(25) **NOT** **NULL**,

dob **date** **NOT** **NULL**,

sex varchar2(25) **NOT** **NULL**,

street varchar2(25) **NOT** **NULL**,

city varchar2(25) **NOT** **NULL**,

postcode varchar2(25) **NOT** **NULL**,

id number(10) **NOT** **NULL**,

program varchar2(25) **NOT** **NULL**,

year number(2),

dept varchar2(5) **NOT** **NULL**,

**CONSTRAINT** pk\_student **PRIMARY** **KEY** (id),

**CONSTRAINT** fk\_student **FOREIGN** **KEY** (id) **REFERENCES** Patron (id) **ON** **DELETE** **CASCADE**,

**CONSTRAINT** fk\_student\_program **FOREIGN** **KEY** (program) **REFERENCES** Degree\_Program (program),

**CONSTRAINT** fk\_student\_dept **FOREIGN** **KEY** (dept) **REFERENCES** Departments (abbreviation)

);

1. **Table Name: Faculty**

Description: Stores the names and basic details of every faculty at NCSU.

Constraints: Primary key constraint is enforced on the id column.

Foreign key constraint is enforced on the following columns:

* id column which references the id column in Patron table
* category column which references the category column in Faculty\_Category table
* dept column which references the abbreviation column in the Departments table

FD: id -> category, id, dept, course\_id

SQL:

**CREATE** **TABLE** Faculty (

category varchar2(25) **NOT** **NULL**,

id number(10) **NOT** **NULL**,

dept varchar2(5) **NOT** **NULL**,

course\_id number(10) **NOT** **NULL**,

**CONSTRAINT** pk\_faculty **PRIMARY** **KEY** (id),

**CONSTRAINT** fk\_faculty **FOREIGN** **KEY** (id) **REFERENCES** Patron (id) **ON** **DELETE** **CASCADE**,

**CONSTRAINT** fk\_faculty\_category **FOREIGN** **KEY** (category) **REFERENCES** Faculty\_Category (category),

**CONSTRAINT** fk\_faculty\_dept **FOREIGN** **KEY** (dept) **REFERENCES** Departments (abbreviation),

**CONSTRAINT** fk\_faculty\_course\_id **FOREIGN** **KEY** (course\_id,dept) **REFERENCES** Courses (id, dep\_abbreviation)

);

1. **Table Name: Reminders**

Description: Stores the messages to be sent to the patron

Constraints: Primary key constraint is enforced on the id column.

Foreign key constraint is enforced on the patron\_id column which references the id column in the patron table

FD: id -> id, message, time\_sent, patron\_id

SQL:

**CREATE** **TABLE** Reminders (

id number(10),

message varchar2(500) **NOT** **NULL**,

time\_sent **date** **NOT** **NULL**,

patron\_id number(20) **NOT** **NULL**,

**CONSTRAINT** pk\_reminders **PRIMARY** **KEY** (id),

**CONSTRAINT** fk\_reminders\_patrons **FOREIGN** **KEY** (patron\_id) **REFERENCES** Patron (id)

);

1. **Table Name: Course\_Taken**

Description: Stores the courses taken by each student and other details about the coursesuch as the year and semester

Constraints: Primary key constraint is enforced on the (patron\_id, dep\_abbreviation, id) columns

Foreign key constraint is enforced on the (id,dep\_abbreviation) columns which references the (id,dep\_abbreviation) column in Courses table

FD: patron\_id, dep\_abbreviation, id -> patron\_id, dep\_abbreviation, id, year, semester

**SQL:**

**CREATE** **TABLE** Course\_Taken (

patron\_id number(10),

dep\_abbreviation varchar2(3),

id number(10),

year number(2),

semester number(2),

**CONSTRAINT** pk\_courses\_taken **PRIMARY** **KEY** (patron\_id, dep\_abbreviation, id),

**CONSTRAINT** fk\_courses\_taken\_1 **FOREIGN** **KEY** (id, dep\_abbreviation) **REFERENCES** Courses (id, dep\_abbreviation),

**CONSTRAINT** fk\_Course\_Taken\_patrons **FOREIGN** **KEY** (id) **REFERENCES** Patron

**ON** **DELETE** **CASCADE**);

1. **Table Name: Authors**

Description: Stores the list of all the names of the authors whose publication(s) is/are present in the NCSU libraries

Constraints: Primary key constraint is enforced on the id column

FD: id -> id, name

SQL:

**CREATE** **TABLE** Authors(

id varchar2(10),

name varchar2(50) **NOT** **NULL**,

**CONSTRAINT** pk\_authors **PRIMARY** **KEY** (id)

);

1. **Table Name: Publications**

Description: Stores the list of all the publications that are present in the NCSU libraries

Constraints: Primary key constraint is enforced on the id column

FD: id -> id, year\_of\_pub, title

SQL:

**CREATE** **TABLE** Publications(

title varchar2(300) **NOT** **NULL**,

id varchar2(10),

year\_of\_pub number(4) **NOT** **NULL**,

**CONSTRAINT** pk\_publications **PRIMARY** **KEY** (id)

);

**ALTER** **TABLE** Publications

MODIFY (year\_of\_pub number(4));---

1. **Table Name: Pub\_ConferencePapers**

Description: Stores the list of all the conference papers (that are actually a subtype of publications) that are present in the NCSU libraries

Constraints: Primary key constraint is enforced on the conf\_num column.

Foreign key constraint is enforced on the conf\_num column which references the id column in the Publications table

FD: conf\_num -> conf\_num, confName

SQL:

**CREATE** **TABLE** Pub\_ConferencePapers(

conf\_num varchar2(10),

confName varchar2(50),

**CONSTRAINT** pk\_pub\_conferencepapers **PRIMARY** **KEY** (conf\_num),

**CONSTRAINT** fk\_pub\_conferencepapers **FOREIGN** **KEY** (conf\_num) **REFERENCES** Publications(id)

**ON** **DELETE** **CASCADE**

);

1. **Table Name: Pub\_Book**

Description: Stores the list of all books (that are actually a subtype of publications) that are present in the NCSU libraries

Constraints: Primary key constraint is enforced on the isbn column.

Foreign key constraint is enforced on the isbn column which references the id column in the Publications table

FD: isbn -> edition, isbn, publisher

SQL:

**CREATE** **TABLE** Pub\_Book(

edition varchar2(5) **NOT** **NULL**,

isbn varchar2(10),

publisher varchar2(50),

**CONSTRAINT** pk\_pub\_book **PRIMARY** **KEY** (isbn),

**CONSTRAINT** fk\_pub\_book **FOREIGN** **KEY** (isbn) **REFERENCES** Publications(id)

**ON** **DELETE** **CASCADE**

);

1. **Table Name: Pub\_Journal**

Description: Stores the list of all journals (that are actually a subtype of publications) that are present in the NCSU libraries

Constraints: Primary key constraint is enforced on the issn column.

Foreign key constraint is enforced on the issn column which references the id column in the Publications table

FD: issn -> issn

SQL:

**CREATE** **TABLE** Pub\_Journal(

issn varchar2(10),

**CONSTRAINT** pk\_pub\_journal **PRIMARY** **KEY** (issn),

**CONSTRAINT** fk\_pub\_journal **FOREIGN** **KEY** (issn) **REFERENCES** Publications(id)

**ON** **DELETE** **CASCADE**

);

1. **Table Name: Written\_by**

Description: Stores the list of publication id and the corresponding author id

Constraints: Primary key constraint is enforced on the (pid,aid) column.

Foreign key constraint is enforced on the following columns:

* aid column which references the id column in Authors table
* pid column which references the id column in Publications table

FD: pid, aid -> pid, aid

SQL:

**CREATE** **TABLE** Written\_by(

pid varchar2(10),

aid varchar2(10),

**CONSTRAINT** fk\_written\_by\_authors **FOREIGN** **KEY** (aid) **REFERENCES** Authors(id),

**CONSTRAINT** fk\_written\_by\_publications **FOREIGN** **KEY** (pid) **REFERENCES** Publications(id),

**CONSTRAINT** pk\_written\_by **PRIMARY** **KEY** (pid,aid)

);

1. **Table Name: Library**

Description: Stores the names of the existing libraries at NCSU

Constraints: Primary key constraint is enforced on the id column.

FD: id -> id, name

SQL

**CREATE** **TABLE** Library(

id number(10),

name varchar2(50),

**CONSTRAINT** pk\_library **PRIMARY** **KEY** (id)

);

1. **Table Name: Copies**

Description: Stores the list of all the copies of each publication that is present in the NCSU libraries and relevant details such as it’s type, status.

Constraints: Primary key constraint is enforced on the id column.

Foreign key constraint is enforced on the following columns:

* lib\_id column which references the id column in Library table
* pid column which references the id column in Publications table.

Check constraint is enforced on the following columns :

* copy\_type to make sure that values being inserted into the column are one of the expected string values : ‘IN’ , ‘ELECTRONIC’ or ‘HARD’
* copy\_status to make sure that values being inserted into the column are one of the expected string values: ‘IN’ or ‘OUT’

FD: id -> id, pid, copy\_type, lib\_id, status

**SQL:**

**CREATE** **TABLE** Copies(

id number(10),

pid varchar2(10),

copy\_type varchar2(25),

lib\_id number(10),

status varchar2(5),

**CONSTRAINT** pk\_copies **PRIMARY** **KEY** (id),

**CONSTRAINT** fk\_copies\_library **FOREIGN** **KEY** (lib\_id) **REFERENCES** Library(id),

**CONSTRAINT** chk\_copy\_type **CHECK**(copy\_type **IN** ('ELECTRONIC','HARD')),

**CONSTRAINT** chk\_copy\_status **CHECK**(status **IN**('IN','OUT')),

**CONSTRAINT** fk\_copy\_pub **FOREIGN** **KEY** (pid) **REFERENCES** Publications(id)

**ON** **DELETE** **CASCADE**

);

1. **Table Name: Reservation**

Description: Stores the list of all reservations (past,present and future) made for a particular copy of a publication by faculty and the necessary details such as start time and end time

Constraints: Primary key constraint is enforced on the (course\_id, copy\_id, dep\_abbreviation, start\_time, end\_time) columns.

Foreign key constraint is enforced on the following columns:

* (course\_id,dep\_abbreviation) columns which references the (id,dep\_abbreviation) columns in Courses table
* copy\_id column which references the id column in Copies table.

FD: course\_id, copy\_id, dep\_abbreviation, start\_time, end\_time -> course\_id, copy\_id, dep\_abbreviation, start\_time, end\_time

**SQL:**

**CREATE** **TABLE** Reservation(

course\_id number(10),

dep\_abbreviation varchar2(3),

copy\_id number(10),

start\_time **date**,

end\_time **date**,

**CONSTRAINT** pk\_reservation **PRIMARY** **KEY** (course\_id, copy\_id, dep\_abbreviation, start\_time, end\_time),

**CONSTRAINT** fk\_reserv\_dep\_abbrev **FOREIGN** **KEY** (course\_id, dep\_abbreviation) **REFERENCES** Courses(id, dep\_abbreviation),

**CONSTRAINT** fk\_reservation\_copy **FOREIGN** **KEY** (copy\_id) **REFERENCES** Copies(id)

);

1. **Table Name: Rooms**

Description: Stores the details of all the rooms, that can be used by a patron, in the NCSU libraries

Constraints: Primary key constraint is enforced on the room\_number column

Foreign key constraint is enforced on the library\_id column which references theid column in Library table

Check constraint is enforced on the room\_type column to make sure tha values being inserted into the column are one of the expected string values: ‘CONF’ or ‘STUDY’

FD: room\_number -> room\_number, capacity, library\_id, floor\_no, room\_type

**SQL:**

**CREATE** **TABLE** Rooms(

room\_number number(10),

capacity number(3) **NOT** **NULL**,

library\_id number(10) **NOT** **NULL**,

floor\_no number(2) **NOT** **NULL**,

room\_type varchar2(25),

**CONSTRAINT** pk\_rooms **PRIMARY** **KEY** (room\_number),

**CONSTRAINT** fk\_rooms\_library **FOREIGN** **KEY** (library\_id) **REFERENCES** Library(id),

**CONSTRAINT** chk\_room\_type **CHECK**(room\_type **IN** ('CONF','STUDY'))

);

1. **Table Name: Cameras**

Description: Stores the details of all the cameras that are owned by the NCSU libraries

Constraints: Primary key constraint is enforced on the id column

Foreign key constraint is enforced on the lid column that references the id column in Library table

FD: id -> id, make, model, config, lid, memory, status

**SQL:**

**CREATE** **TABLE** Cameras(

id number(10),

make varchar2(20),

model VARCHAR2(20) **NOT** **NULL**,

config VARCHAR2(50),

lid number(1),

memory VARCHAR2(20),

status number(1),

**CONSTRAINT** pk\_cameras **PRIMARY** **KEY** (id),

**CONSTRAINT** fk\_library **FOREIGN** **KEY** (lid) **REFERENCES** Library(id)

);

1. **Table Name: CHECKS\_OUT**

Description: Stores the details of all the copies that were checked out (past, present and future) by a patron

Constraints: Primary key constraint is enforced on the id column

Foreign key constraint is enforced on the copy\_id column that references the id column in Copies table

FD: id -> id, patron\_id, copy\_id, start\_time, end\_time

**SQL:**

**CREATE** **TABLE** CHECKS\_OUT(

id number(10),

patron\_id number(10),

copy\_id number(10),

start\_time **date**,

end\_time **date**,

**CONSTRAINT** pk\_checks\_out **PRIMARY** **KEY** (id),

**CONSTRAINT** fk\_checks\_out\_patron **FOREIGN** **KEY** (patron\_id) **REFERENCES** Patron(id),

**CONSTRAINT** fk\_checks\_out\_copies **FOREIGN** **KEY** (copy\_id) **REFERENCES** Copies(id)

);

**alter** **table** checks\_out

**add** ( ACT\_RETURN\_TIME **date**);---

1. **Table Name: Fines**

Description: Stores the details of individual (for each activity of checking out and failing to return a copy of a particular publication on time) fines incurred by a patron.

Constraints: Primary key constraint is enforced on the id column

Foreign key constraint is enforced on the checks\_out\_id column that references the id column in CHECKS\_OUT table

Check constraint is enforced on the status column to make sure that the values being inserted are one of the expected string values: ‘PAID’ or ‘UNPAID’

FD: id -> id, checks\_out\_id, amount, status, fine\_date

**SQL:**

**CREATE** **TABLE** Fines (

id number(10),

checks\_out\_id number(10),

amount number(10) **NOT** **NULL**,

status varchar2(25),

fine\_date **date**,

**CONSTRAINT** pk\_fines **PRIMARY** **KEY** (id),

**CONSTRAINT** fk\_fines\_checks\_out **FOREIGN** **KEY** (checks\_out\_id) **REFERENCES** CHECKS\_OUT(id),

**CONSTRAINT** chk\_fine\_status **CHECK**(status **IN** ('PAID','UNPAID'))

);

1. **Table Name: Booked**

Description: Stores the details of all reservations made for each room in the NCSU libraries by a patron

Constraints: Primary key constraint is enforced on the (room\_number, start\_time, end\_time) columns

Foreign key constraint is enforced on the following columns:

* patron\_id column which references id column in Patron table
* room\_number column which references room\_number column in Rooms table

Check constraint is enforced on the following columns:

* end\_time and start\_time- to make sure that end time being inserted is greater than the start time being inserted
* status column to make sure that values being inserted into the column are one of the expected values: ‘VALID’ or ‘INVALID’

FD: room\_number, start\_time, end\_time -> patron\_id, room\_number, start\_time, end\_time, checked\_out, checked\_in, status

**SQL:**

**CREATE** **TABLE** Booked(

patron\_id number(10),

room\_number number(10),

start\_time **date**,

end\_time **date**,

checked\_out **date**,

checked\_in **date**,

status varchar2(7),

**CONSTRAINT** pk\_booked **PRIMARY** **KEY** (room\_number, start\_time, end\_time),

**CONSTRAINT** fk\_booked\_patron **FOREIGN** **KEY** (patron\_id) **REFERENCES** Patron(id),

**CONSTRAINT** fk\_booked\_rooms **FOREIGN** **KEY** (room\_number) **REFERENCES** Rooms(room\_number),

**CONSTRAINT** chk\_start\_end **CHECK**(end\_time > start\_time),

**CONSTRAINT** chk\_booking\_status **CHECK**(status **IN**('VALID','INVALID'))

);

1. **Table Name: Booked\_Cams**

Description: Stores the details of all reservations made for each camera (owned by the NCSU libraries) by a patron

Constraints: Primary key constraint is enforced on the id column.

Unique key constraint is enforced on the (cam\_id, patron\_id, start\_time, end\_time)

Foreign key constraint is enforced on the following columns:

* patron\_id column which references id column in Patron table
* cam\_id column which references id column in Cameras table

FD: id -> id, cam\_id, patron\_id, start\_time, end\_time

cam\_id, patron\_id, start\_time, end\_time -> id, cam\_id, patron\_id, start\_time, end\_time

SQL:

**CREATE** **TABLE** Booked\_Cams(

id number(10),

cam\_id number(10),

patron\_id number(10),

start\_time **date**,

end\_time **date**,

**CONSTRAINT** pk\_booked\_cams **PRIMARY** **KEY** (id),

**CONSTRAINT** fk\_booked\_cams\_patron **FOREIGN** **KEY** (patron\_id) **REFERENCES** Patron(id),

**CONSTRAINT** fk\_booked\_cams\_cameras **FOREIGN** **KEY** (cam\_id) **REFERENCES** Cameras(id),

**CONSTRAINT** uk\_booked\_cams **UNIQUE** (cam\_id, patron\_id, start\_time, end\_time)

);

1. **Table Name: Cam\_Fines**

Description: Stores the details of individual (for each activity of checking out and failing to return a camera on time) fines incurred by a patron.

Constraints: Primary key constraint is enforced on the id column

Foreign key constraint is enforced on the booked\_cam\_id column that references the id column in Booked\_Cams table

Check constraint is enforced on the status column to make sure that the values being inserted are one of the expected string values: ‘PAID’ or ‘UNPAID’

FD: id -> id, booked\_cam\_id, amount, status

SQL:

**CREATE** **TABLE** Cam\_Fines (

id number(10),

booked\_cam\_id number(10),

amount number(10) **NOT** **NULL**,

status varchar2(25),

**CONSTRAINT** pk\_cam\_fines **PRIMARY** **KEY** (id),

**CONSTRAINT** fk\_fines\_booked\_cams **FOREIGN** **KEY** (booked\_cam\_id) **REFERENCES** Booked\_Cams(id),

**CONSTRAINT** chk\_cam\_fine\_status **CHECK**(status **IN** ('PAID','UNPAID'))

);

1. **Normalisation:**

All the table listed are in BCNF, which is the ideal normalization form.

1. **Procedures, Functions and Schedulers:**

All the procedures and functions are stored in the “Procedures\_Functions” folder in the zipfile. Schedulers are stored in the “Schedulers” folder.

1. **Views:**

All the views implemented are stored in the “Views” folder in the zipfile

1. **Other Constraints:**
2. Constraints implemented in the application side i.e in Java
3. All cameras have a check out duration of Friday till Thursday of the following week. Check out time is between 9 am noon am on Fridays. This constraint has been implemented in Java code i.e application side. The reason for implementing this in the application side was that we wanted to completely disallow (i.e we did not want the user to try to checkout during an unacceptable checkout time and get an error. We simply wanted to stop any attempts when he tried such an action ) the user from trying to check out the camera during a time of his choice. We could implement this only in the application side.
4. Good or Bad status maintenance for the patron (depending on whether he has cleared fines or not). Table constraints cannot adequately implement the required constraints for status maintenance. While this could have been implemented using procedures (As part of the database) this would have required us to enforce the constraint in multiple places, making it quite difficult to implement. Thus we chose to implement this in the application side.
5. Constraints implemented in the database i.e through Procedures. These constraints were implemented through procedures because they cannot be adequately implemented with the features available in table definitions. The constraints are as follows:
6. Based on the classification of a student (for example if a student is an undergraduate) he may or may not be allowed to be in a particular degree program (undergraduate student cannot pursue a Phd)
7. Electronic copies cannot be checked out i.e no return date or check out date and hence they will always be available.
8. If a copy is reserved by a faculty- only students who are taking the course taught by that particular faculty can take that copy and for fours hours only.
9. If a copy is checked out by a student and another student tries to check out the same copy, he is placed in a queue. Faculty can also try to checkout a copy that has already been checked out and they will be given higher preference in the queue over other students
10. After four months of reservation: book automatically unreserved.
11. If a patron returns a publication and someone is in the queue, the book is automatically checked out for that user
12. If a publication is not returned before end time, fines are automatically charged to the patron and are incremented with each day he goes without returning the copy
13. If there are patrons waiting in the queue for a particular copy, the person who is currently in possession of the copy will not be allowed to renew the copy i.e to extend the time.
14. **SQL Queries and Snapshots of the Result** (Solution in same order as the queries posted in the document)**:**
15. **CREATE** **OR** REPLACE **VIEW** QUERY1 **AS**

(**SELECT** t.FNAME **as** FNAME, t.LNAME **as** LNAME, SUM(TEMP) **AS** "NUMBER OF TIMES FAILED" **FROM**

(**SELECT** P.FNAME, P.LNAME,1 **AS** TEMP

**FROM** PATRON P, CHECKS\_OUT C, COPIES\_PUB\_SCREEN\_VIEW PUB

**WHERE** PUB.ID = C.COPY\_ID **AND** P.ID=C.PATRON\_ID **AND** (C.ACT\_RETURN\_TIME>C.END\_TIME **OR** (C.END\_TIME<SYSDATE **AND** C.ACT\_RETURN\_TIME **IS** **NULL**)) **AND** C.END\_TIME>='05-OCT-15' **AND** C.END\_TIME<='05-NOV-15'

**UNION** **ALL**

**SELECT** P.FNAME, P.LNAME,1 **AS** TEMP **FROM** PATRON P, BOOKED B

**WHERE** ((B.STATUS = 'VALID' **AND** B.CHECKED\_IN>B.END\_TIME) **OR**

(B.STATUS = 'VALID' **AND** B.CHECKED\_IN **IS** **NULL** **AND** B.END\_TIME<SYSDATE)) **AND**

B.PATRON\_ID = P.ID

**UNION** **ALL**

**SELECT** P.FNAME, P.LNAME,1 **AS** TEMP **FROM** PATRON P, BOOKED\_CAMS C

**WHERE** ((C.RETURNED\_TIME>C.END\_TIME) **OR**

(C.RETURNED\_TIME **IS** **NULL** **AND** C.END\_TIME<SYSDATE)) **AND**

C.PATRON\_ID = P.ID) t **GROUP** **BY** t.FNAME, t.LNAME ) **ORDER** **BY** FNAME **DESC**;---

Result:



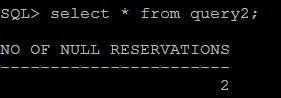
1. **CREATE** **OR** REPLACE **VIEW** QUERY2 **AS**

(**SELECT** COUNT(\*) **AS** "NO OF NULL RESERVATIONS"

**FROM** PATRON P, BOOKED B

**WHERE** P.ID=B.PATRON\_ID **AND** B.CHECKED\_OUT **IS** **NULL** **AND** B.CHECKED\_IN **IS** **NULL** **AND** B.STATUS='INVALID' **AND** B.END\_TIME>='30-OCT-15' **AND** B.END\_TIME<='06-NOV-15');---

Result:



1. **CREATE** **OR** REPLACE **VIEW** QUERY3 **AS**

(**SELECT** SUM(t.TOTAL) **as** TOTAL\_FINES

**FROM**

(**SELECT** SUM(AMOUNT) **AS** TOTAL

**FROM** FINES F, CHECKS\_OUT C

**WHERE** F.CHECKS\_OUT\_ID=C.ID **AND** F.FINE\_DATE>='05-OCT-15' **AND** F.FINE\_DATE<='05-NOV-15'

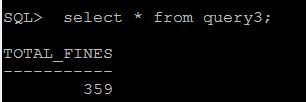
**UNION**

**SELECT** SUM(AMOUNT) **AS** TOTAL

**FROM** CAM\_FINES CF, BOOKED\_CAMS BC

**WHERE** CF.BOOKED\_CAM\_ID=BC.ID **AND** BC.END\_TIME>='05-OCT-15' **AND** BC.END\_TIME<='05-NOV-15') t);---

Result:



1. **CREATE** **OR** REPLACE **VIEW** QUERY4 **AS**

(**SELECT** t.CAPACITY,t.NO\_OF\_REQUESTS

**FROM** (

**SELECT** R.CAPACITY, COUNT(\*) **AS** "NO\_OF\_REQUESTS"

**FROM** BOOKED B, ROOMS R, PATRON P

**WHERE** R.ROOM\_NUMBER=B.ROOM\_NUMBER **AND** B.PATRON\_ID=P.ID **AND** B.START\_TIME>='05-OCT-15' **AND** B.START\_TIME<='05-NOV-15'

**GROUP** **BY** CAPACITY **ORDER** **BY** NO\_OF\_REQUESTS **DESC**) t

**WHERE** ROWNUM <= 1);---

Result:

