UCS310: Information Management System

Project on

Time Table

Management System

Submitted By:

Madhav Goyal: 101610050

Madanjeet kumar: 101603176

(COE 13)

2017-18

Submitted To:

Ms. Avleen Kaur Malhi

Assistant Professor

(CSED)

CERTIFICATE

This is to certify the project entitled “Time management System” is an original work carried under my supervision of in particular fullfilment for the award of the degree of bachelor of engineering from Thapar University ,Patiala ,Punjab during the month of Dec 2017.

The project report has been approved by me as, it satisfies the academic requirement in the respect of project work prescribed for the bachelor of degree .

Ms. Avleen Kaur

ACKNOWLEDGEMENT

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and teachers. We would like to extend our sincere thanks to all of them.

We are highly indebted to Mrs. Avleen kaur their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

We would like to express our gratitude towards our parents for their kind co-operation and encouragement which helped us in completion of this project.

Our thanks and appreciations also go to our colleagues in developing the project and people who have willingly helped us out with their abilities.

INTRODUCTION

Technology makes lifestyle easier by providing better support to different systems, better accuracy , better security options , easier maintenance, etc. Now a day's technology eventually means "computers" which is the greatest achievements of last century. Day by day computers are being more and more popular because of its features like ease of work, ease of learning , greater accuracy with the least time consumption and the last but not the least i.e. ease of maintenance with cost effectiveness .

So as a part of these ongoing evolutionary approach traditional systems are being computerized to make them more fruitful than ever.

ABSTRACT

Time Table Management system is an automated system which genets time table according to the data given by the user. The main requirement of the application is to provide the details about the branch, subjects, no. of labs , total no. of period and details about the lab assistance .Then the application generates the time table according to your need.

PROJECT DEFINITION

- The basic project is to create a **Time Table Management System**
- To create Databases of different entities involved in this process.
- Maintaining database - containing information about the various semesters , subjects , labs , teachers etc.

NEED

- As we discussed earlier that manual maintainance of a Timetable Management of a TimeTable Management System is a tedious job. So to enhance the ease of working we go for this package .
- Manual maintenance of databases of items, time table processing is a time taking process and somehow erroneous.

- To give more accuracy to the system i.e. rather going manual modification we involve computer for accuracy.

- The least but most important it saves time.

PROJECT OVERVIEW

Objectives of the package

- Create a Time Table Management System to be used by any College.
- To perform the basic requirements of the firm.
- Maintaining databases of subject, Class, semester's details.

Scopes and boundaries of the package

- As it is a computer-based package so maintenance and working is somewhat difficult from manual mode of approach.
- As it is not possible to associate each and every requirement of the system so in some way or other it will going to create problem at some stage of execution (like report generation).
- As a computer based System it is easier to fetch data from the database for unsocial activities. Also easier to destroy the existing ones.

Expected Benefits

- On implementing this package the farm will get error free data to analyze.
- This package would limit the time and money factor involve in "Time Table Management System".
- Maintenance is much easier and accurate than the existing manual system.
- Security features are somewhat higher than that of manual approach.
- This software helps to prepare the timetable in universities/colleges. It is equipped table in universities/colleges. It is equipped with built in time table creation engine ,with built in time table creation engine ,which will create the time table which will create the timetable automatically and within a short period of automatically and within a short period of time based on the instruction provided by time based on the instruction provided by the user. The time table created will be of the user. The time table created will be of very good accuracy than the manual very good

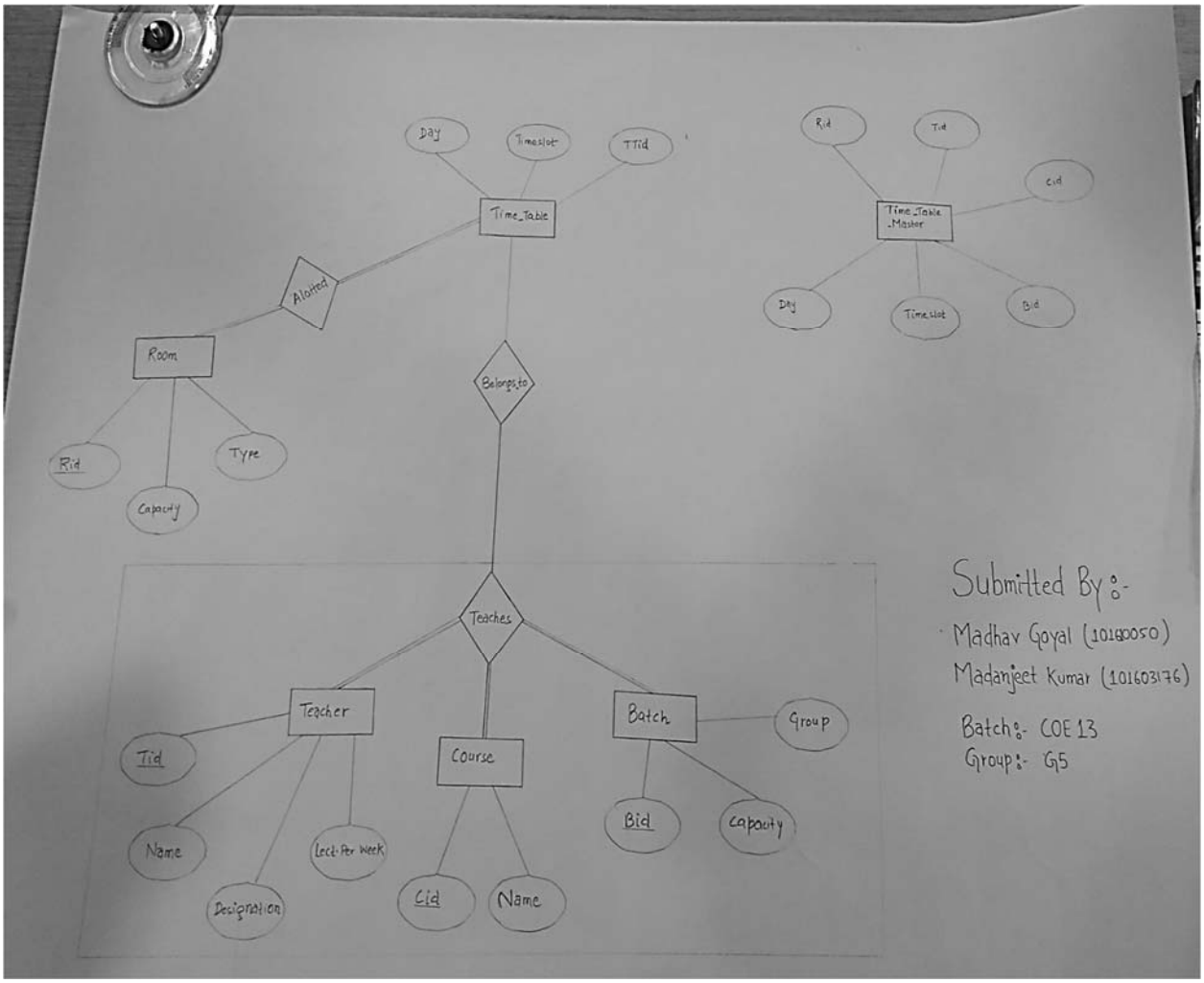
accuracy than the manual method. One can retain the same or can method. One can retain the same or can make some changes to it and retain .make some changes to it and retain.

Hardware and Software Configuration

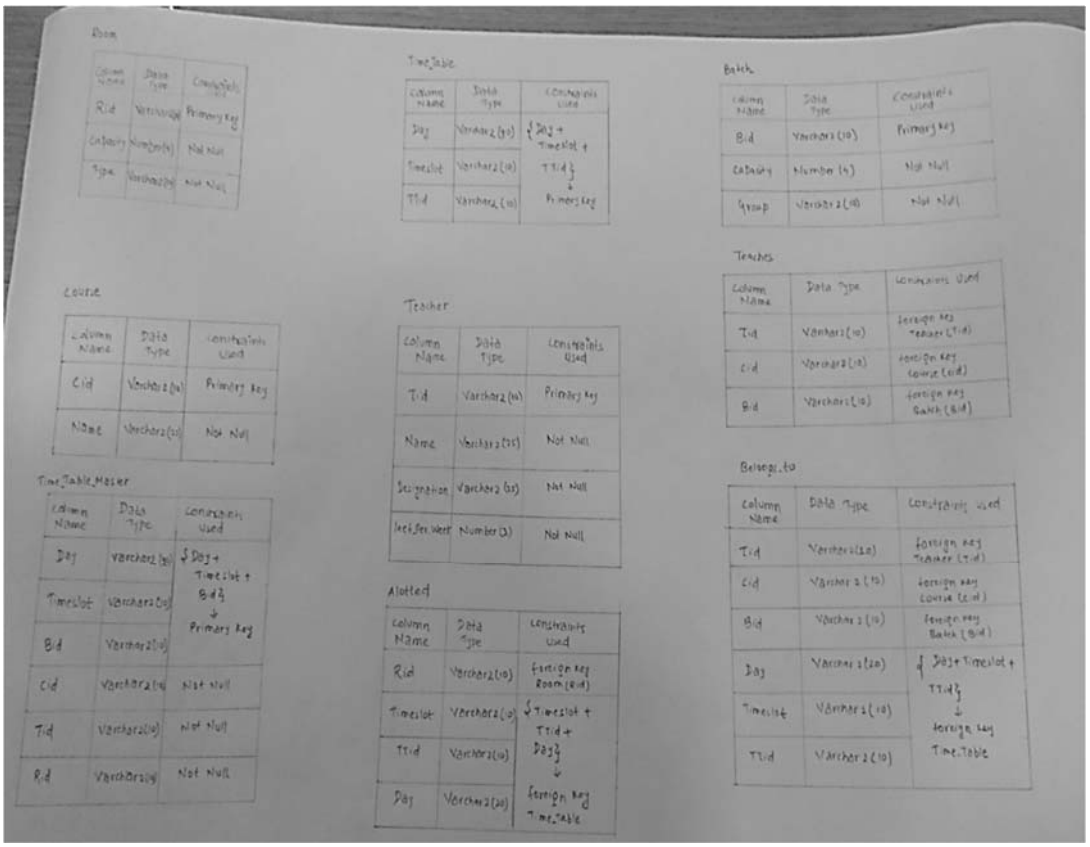
The hardware and software should be chosen carefully keeping following point in mind

- The System must be user friendly.
- The System must be able to handle large number of data.
- Processing speed of the system must be fast.

ER DIAGRAM


Day
Timeslot
TTid
Time_Table
Allotted
Room
Rid
Type
Capacity
Belongs to
Rid
Tid
Cid
Time_Table_Master
Day
Timeslot
Bid
Teaches
Teacher
Tid
Name
Designation
Lect-per week
Course
Cid
Name
Batch
Group
Bid
capacity
Submitted By :-
Madhav Goyal (101803050)
Madanjeet Kumar (101603176)
Batch :- COE 13
Group :- G5

TABLES BEFORE NORMALIZATION



Room

Column Name	Data Type	Constraints Used
Rid	Varchar2(10)	Primary key
Capacity	Number(4)	Not Null
Type	Varchar2(5)	Not Null

TimeTable

Column Name	Data Type	Constraints Used
Day	Varchar2(10)	{Day + Timeslot + TTid} → Primary key
Timeslot	Varchar2(10)	
TTid	Varchar2(10)	

Batch

Column Name	Data Type	Constraints Used
Bid	Varchar2(10)	Primary key
Capacity	Number(4)	Not Null
Group	Varchar2(10)	Not Null

Course

Column Name	Data Type	Constraints Used
Cid	Varchar2(10)	Primary key
Name	Varchar2(20)	Not Null

Teacher

Column Name	Data Type	Constraints Used
Tid	Varchar2(10)	Primary key
Name	Varchar2(25)	Not Null
Designation	Varchar2(15)	Not Null
Teaching Hours	Number(2)	Not Null

Teaches

Column Name	Data Type	Constraints Used
Tid	Varchar2(10)	foreign key Teacher(Tid)
Cid	Varchar2(10)	foreign key Course(Cid)
Bid	Varchar2(10)	foreign key Batch(Bid)

TimeTable_Master

Column Name	Data Type	Constraints Used
Day	Varchar2(10)	{Day + Timeslot + Bid} → Primary key
Timeslot	Varchar2(10)	
Bid	Varchar2(10)	
Cid	Varchar2(10)	Not Null
Tid	Varchar2(10)	Not Null
Rid	Varchar2(10)	Not Null

Allotted

Column Name	Data Type	Constraints Used
Rid	Varchar2(10)	foreign key Room(Rid)
Timeslot	Varchar2(10)	{Timeslot + TTid + Day} → foreign key Time_Table
TTid	Varchar2(10)	
Day	Varchar2(20)	

Belongs to

Column Name	Data Type	Constraints Used
Tid	Varchar2(10)	foreign key Teacher(Tid)
Cid	Varchar2(10)	foreign key Course(Cid)
Bid	Varchar2(10)	foreign key Batch(Bid)
Day	Varchar2(20)	{Day + Timeslot + TTid} → foreign key Time_Table
Timeslot	Varchar2(10)	
TTid	Varchar2(10)	

Why you need database normalization?

What is Normalization?

Normalization is the procedure to split the relation into relations with less attributes thereby minimizing the redundancy of the data and minimizing the insertions, deletions and updating.

There is a sequence of stages or steps on which the normalization works. This sequence is called normal forms. The normal forms are relevant to entity relations. If the form satisfies a certain sets of constraints then a table or a relation comes to the particular normal form. There are above five normal forms. These are 1NF, 2NF, 3NF, 4NF and 5NF. Here NF stands Normal Form. It is very important for relational data model to consider that the first normal form (1NF) is very grave in creating the relations. Rests of all the forms are optional. There are some guidelines which help to create a good database. There is always a need for updating the database, so that there are three anomalies for data modification.

- **Insert**–This irregularity refers to the circumstances when one cannot insert a new row (tuple) into a relation because of the shortage of data.

- **Delete**-The delete anomaly refers to a state of affairs where the removal of records outcome in accidental loss of several other significant data.

- **Update**-The update abnormality is a state wherever an update of a lone data value needs numerous rows of data to be updated.

Advantages of normalization are as follows-

- Reduce the redundancy.
- A large amount added bendable database design.
- Better overall database association.
- Data uniformity inside the database.

TABLES AFTER NORMALIZATION

In our table ,only one tables are taking participate in normalization i.e **"TIME_TABLE_MASTER".** Means except **TIME_TABLE_MASTER** table ,remaining all tables will be same .Therefore, we have to normalize these above two table and after normalization these two table will split into four table. These four tables are:

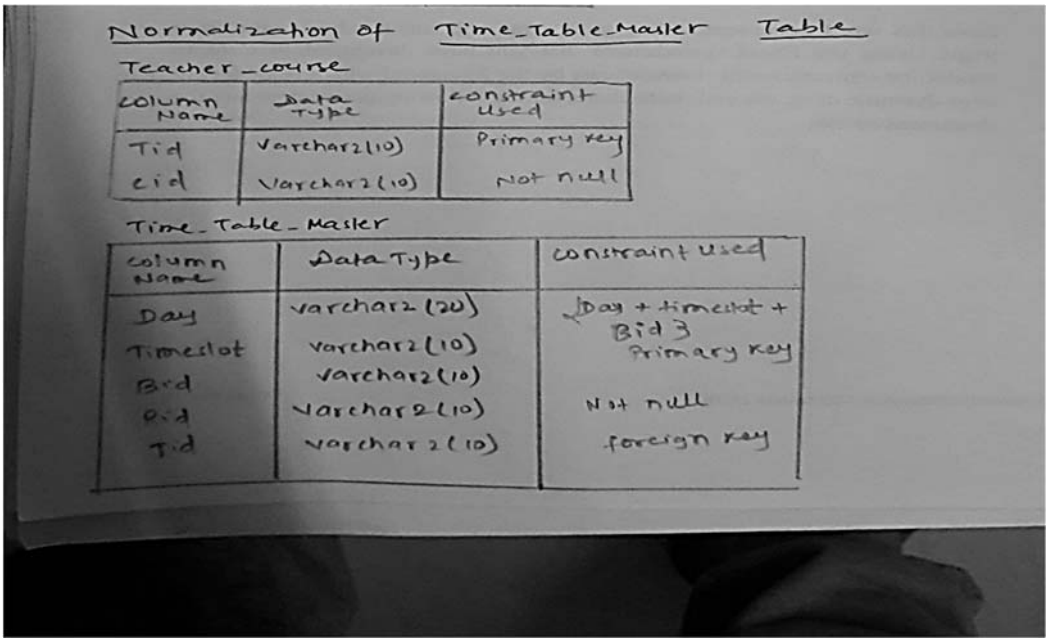


Table Functional Dependencies

Identified Functional Dependencies

TABLE: TEACHER

Tid → Name

Tid → Designationsss

Tid → Lect_per_week

TABLE: ROOM

Rid → Capacity

Rid → Type

TABLE: COURSE

Cid → Name

TABLE: BATCH_GROUP

Bid → Group

TABLE: GROUP_CAPACITY

Group → Capacity

→ →

TABLE: TIME_TABLE_MASTER

(Daily + Timeslot + Bid) Rid

(Daily + Timeslot + Bid) Tid

TABLE: TEACHER_COURSE

Tid cid

Normalized form of Tables

Keeping in mind the points discussed above, certain normalizations can be made in the given tables. Following changes have been made after normalization of the tables, hence getting the final table structure of the Database.

- ROOM table is already normalized by nature
- COURSE table is already normalized by nature
- TEACHER table is a normalization table
- TIME_TABLE table is a normalization table
- BUT we have to normalize “BATCH” table into “BATCH_GROUP” AND “GROUP_CAPACITY” table.
- And finally we have to also normalize “TIME_TABLE_MASTER” table into “TIME_TABLE_MASTER” table AND “TEACHER_COURSE” table.

CREATION OF TABLE

```
CREATE TABLE  "ROOM"
  (       "RID" VARCHAR2(10),
          "CAPACITY" NUMBER(4,0) NOT NULL ,
          "TYPE1" VARCHAR2(20) NOT NULL ,
           PRIMARY KEY ("RID")
  );

  CREATE TABLE  "TIME_TABLE"
  (       "DAY" VARCHAR2(20),
          "TIMESLOT" VARCHAR2(20),
          "TTID" VARCHAR2(10),
           PRIMARY KEY ("DAY", "TIMESLOT", "TTID")
  );

  CREATE TABLE  "ALLOTTED"
  (       "RID" VARCHAR2(10) REFERENCES  "ROOM" ("RID"),
          "TIMESLOT" VARCHAR2(20),
          "TTID" VARCHAR2(10),
          "DAY" VARCHAR2(20)
          FOREIGN KEY ("DAY", "TIMESLOT", "TTID") REFERENCES  "TIME_TABLE"
("DAY", "TIMESLOT", "TTID") ON DELETE CASCADE
  );

  CREATE TABLE  "BATCH"
  (       "BID" VARCHAR2(10),
          "GROUP1" VARCHAR2(10) NOT NULL ,
          "CAPACITY" NUMBER(3,0) NOT NULL ,
           PRIMARY KEY ("BID")
```

```
);

CREATE TABLE  "COURSE"
(       "CID" VARCHAR2(10),
        "NAME" VARCHAR2(25) NOT NULL ,
         PRIMARY KEY ("CID")
);

CREATE TABLE  "TEACHER"
(       "TID" VARCHAR2(10),
        "NAME" VARCHAR2(25) NOT NULL ,
        "DESIGNATION" VARCHAR2(30) NOT NULL ,
        "LECT_PER_WEEK" NUMBER(2,0) NOT NULL ,
         PRIMARY KEY ("TID")
);

CREATE TABLE  "BELONGS_TO"
(       "TID" VARCHAR2(10) REFERENCES  "TEACHER" ("TID") ON DELETE SET
NULL,
        "CID" VARCHAR2(10) REFERENCES  "COURSE" ("CID") ON DELETE
CASCADE,
        "BID" VARCHAR2(10) REFERENCES  "BATCH" ("BID") ON DELETE
CASCADE,
        "DAY" VARCHAR2(20),
        "TIMESLOT" VARCHAR2(20),
        "TTID" VARCHAR2(10),
        FOREIGN KEY ("DAY", "TIMESLOT", "TTID") REFERENCES  "TIME_TABLE"
("DAY", "TIMESLOT", "TTID") ON DELETE CASCADE
);

CREATE TABLE  "TEACHER_COURSE"
(       "TID" VARCHAR2(10),
```

```
        "CID" VARCHAR2(10) NOT NULL ,
         PRIMARY KEY ("TID")
  );

  CREATE TABLE  "TEACHES"
  (       "TID" VARCHAR2(10) REFERENCES  "TEACHER" ("TID") ON DELETE SET
NULL,
        "CID" VARCHAR2(10) REFERENCES  "COURSE" ("CID") ON DELETE
CASCADE,
        "BID" VARCHAR2(10) REFERENCES  "BATCH" ("BID") ON DELETE
CASCADE
  );

  CREATE TABLE  "TIME_TABLE_MASTER"
  (       "DAY" VARCHAR2(20),
        "TIMESLOT" VARCHAR2(20),
        "BID" VARCHAR2(10),
        "RID" VARCHAR2(10) NOT NULL ,
        "TID" VARCHAR2(10) NOT NULL REFERENCES  "TEACHER_COURSE"
("TID"),
         PRIMARY KEY ("DAY", "TIMESLOT", "BID")
  );
```

CREATION OF PROCEDURE

```
CREATE OR REPLACE PROCEDURE  "RESET_TIME_TABLE" (day1 in varchar2) is

begin
delete from time_table where day=day1;

insert into time_table select day, timeslot, bid from time_table_master where day=day1;

end;
/
/
CREATE OR REPLACE PROCEDURE  "RESET_ALLOTTED" (day1 in varchar2) is

begin
delete from allotted where day=day1;

insert into allotted select rid, timeslot, bid, day from time_table_master where day=day1;

end;
/
/
CREATE OR REPLACE PROCEDURE  "RELEASE_ROOM" (rid1 in varchar2, timeslot1
in varchar2, ttid1 in varchar2, day1 in varchar2) is
rid2 varchar2(10);

begin
select rid into rid2 from allotted where rid=rid1 and timeslot=timeslot1 and ttid=ttid1 and
day=day1;
```

```
delete from allotted where rid=rid1 and timeslot=timeslot1 and ttid=ttid1 and day=day1;

delete from time_table where day=day1 and timeslot=timeslot1 and ttid=ttid1;

dbms_output.put_line('ROOM RELEASED');


exception

when no_data_found then

dbms_output.put_line('ROOM ALREADY VACANT');


end;

/

/

CREATE OR REPLACE PROCEDURE  "INSERT_TIME_TABLE_MASTER" (day in
varchar2, timeslot in varchar2, bid in varchar2, cid in varchar2, tid in varchar2, rid in
varchar2) is


begin

insert into time_table_master values(day, timeslot, bid, rid, tid);

insert into teacher_course values(tid, cid);


end;

/

/

CREATE OR REPLACE PROCEDURE  "INSERT_TEACHES" (tid in varchar2, cid in
varchar2, bid in varchar2) is


begin

insert into teaches values(tid, cid, bid);


end;

/

/
```

```
CREATE OR REPLACE PROCEDURE  "INSERT_TEACHER" (tid in varchar2, name in varchar2, designation in varchar2, lect in number) is


begin

insert into teacher values(tid, name, designation, lect);


end;

/

/

CREATE OR REPLACE PROCEDURE  "INSERT_ROOM" (rid in varchar2, capacity in number, type1 in varchar2) is

begin

   insert into room values(rid, capacity, type1);

end;

/

/

CREATE OR REPLACE PROCEDURE  "INSERT_COURSE" (cid in varchar2, name in varchar2) is


begin

insert into course values(cid, name);


end;

/

/

CREATE OR REPLACE PROCEDURE  "INSERT_BELONGS" (tid in varchar2, cid in varchar2, bid in varchar2, day1 in varchar2, timeslot in varchar2, ttid in varchar2) is


begin

insert into belongs_to values(tid, cid, bid, day1, timeslot, ttid);


end;
```

```
/

/

CREATE OR REPLACE PROCEDURE  "INSERT_BATCH" (bid in varchar2, group1 in
varchar2, capacity1 in number) is


begin

insert into batch values(bid, group1,capacity1);


end;

/

/

CREATE OR REPLACE PROCEDURE  "CHECK_ROOM" (choice in number, timeslot1 in
varchar2, day1 in varchar2, rid1 in varchar2, type1 in varchar2, capacity1 in number) as

rid2 varchar2(10);


begin

if choice=1 then

select rid into rid2 from allotted where rid=rid1 and timeslot=timeslot1 and day=day1;

dbms_output.put_line('ROOM IS OCCUPIED');


elsif choice=2 then

select a.rid into rid2 from allotted a, room r where (a.timeslot=timeslot1 and a.day=day1) and
(r.capacity=capacity1 and r.type1=type1) and a.rid=r.rid;

dbms_output.put_line('ROOM IS OCCUPIED');


end if;


exception

when no_data_found then

dbms_output.put_line('ROOM IS VACANT');
```

```
end;
/
/
CREATE OR REPLACE PROCEDURE  "BOOK_ROOM" (rid1 in varchar2, timeslot1 in
varchar2, ttid1 in varchar2, day1 in varchar2) is
rid2 varchar2(10);

begin
select rid into rid2 from allotted where rid=rid1 and timeslot=timeslot1 and ttid=ttid1 and
day=day1;
raise_application_error(-20001, 'ROOM ALREADY BOOKED');

exception
when no_data_found then
insert into allotted values(rid1, timeslot1, ttid1, day1);
insert into time_table values(day1, timeslot1, ttid1);
dbms_output.put_line('ROOM BOOKED');

end;
/
/
```

CREATION OF TRIGGER

```
CREATE OR REPLACE TRIGGER  "INSERT_TIME_TABLE" after insert on
time_table_master
for each row
declare
bid1 time_table_master.bid%type;
begin
select bid into bid1 from time_table_master where bid=:new.bid;
insert into time_table values(:new.day, :new.timeslot, :new.bid);
exception
when no_data_found then
null;
end;
/
```

```
CREATE OR REPLACE TRIGGER  "INSERT_ALLOTTED" after insert on
time_table_master
for each row
declare
rid1 time_table_master.rid%type;
begin
select rid into rid1 from time_table_master where rid=:new.rid;
insert into allotted values(:new.rid, :new.timeslot, :new.bid, :new.day);
exception
when no_data_found then
null;
end;
/
```

CURSORS

```
/*################cusror - vacant_room#################*/
declare
        timeslot2 varchar2(10);
        day2 varchar2(20);

        cursor vacant_room(timeslot1 varchar2, day1 varchar2) is select rid from room minus
select rid from allotted where timeslot=timeslot1 and day=day1;

begin
        timeslot2 := &timeslot;
        day2 := &day1;

        for rid1 in vacant_room(timeslot2, day2) loop
                dbms_output.put_line(rid1);
        end loop;
end;
/*################################################*/



/*###################time_table_student#####################*/
DECLARE
        name_c varchar2(25);
        name_t varchar2(25);
        rid1 varchar2(10);
        day2 varchar2(20);
        bid2 varchar2(10);
```

```
	cursor time_table_student(day1 varchar2, bid1 varchar2) is select tid, cid, timeslot
from belongs_to where day=day1 and bid=bid1;
BEGIN
	day2 := &day;
	bid2 := &bid;


	for rec in time_table_student(day2, bid2) loop
		select name into name_c from course where cid=rec.cid;
		select name into name_t from teacher where tid=rec.tid;
		select rid into rid1 from allotted where timeslot=rec.timeslot and ttid=bid2 and
day=day2;
		dbms_output.put_line('TIMESLOT: ' || rec.timeslot);
		dbms_output.put_line('COURSE ID: ' || rec.cid);
		dbms_output.put_line('COURSE NAME: ' || name_c);
		dbms_output.put_line('TEACHER ID: ' || rec.tid);
		dbms_output.put_line('TEACHER NAME: ' || name_t);
		dbms_output.put_line('ROOM: ' || rid1);
		dbms_output.put_line(' ');
	end loop;
END;
/*##############################################################*/




/*###################time_table_teacher#####################*/
DECLARE
	name_c varchar2(25);
	rid1 varchar2(10);
	day2 varchar2(20);
	tid2 varchar2(10);
```

```
	cursor time_table_teacher(day1 varchar2, tid1 varchar2) is select bid, cid, timeslot
from belongs_to where day=day1 and tid=tid1;
BEGIN
	day2 := &day;
	tid2 := &tid;


	for rec in time_table_teacher(day2, tid2) loop
		select name into name_c from course where cid=rec.cid;
		select rid into rid1 from allotted where timeslot=rec.timeslot and ttid=rec.bid
and day=day2;
		dbms_output.put_line('TIMESLOT: ' || rec.timeslot);
		dbms_output.put_line('BATCH ID: ' || rec.bid);
		dbms_output.put_line('COURSE ID: ' || rec.cid);
		dbms_output.put_line('COURSE NAME: ' || name_c);
		dbms_output.put_line('ROOM: ' || rid1);
		dbms_output.put_line(' ');
	end loop;
END;
/*##############################################################*/
```