

Hostel Management System

A Project Under Prof. Shashikiran V

Project Review:

- Priyanshu Gupta (18BIT0146)
- Devang Panwar (18BIT0018)
- Ojasvi (18BIT0029)



VIT[®]
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF INFORMATION TECHNOLOGY AND ENGINEERING
VELLORE INSTITUTE OF TECHNOLOGY

OCTOBER 2019

ABSTRACT

“HOSTEL MANAGEMENT SYSTEM” is software developed for managing various activities in the hostel. For the past few years the number of educational institutions is increasing rapidly. Thereby the number of hostels is also increasing for the accommodation of the students studying in this institution. And hence there is a lot of strain on the persons who are running the hostels and software are not usually used in this context. This particular project deals with the problems on managing a hostel and avoids the problems which occur when carried manually. Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system with the system which is more user friendly and more GUI oriented. We can improve the efficiency of the system, thus overcome the drawbacks of the existing system.

INDEX

1. Description of the mini world scenario chosen by the PROJECT TEAM
2. Functional Requirements and Conceptual view of the Database Design
3. Relational schema with sample extension.
4. Relevant SQL DDL, DML statements with constraints.
5. Stored procedures.
6. Triggers and conclusion.

Introduction

In our current era of automated systems with it being either software or hardware, it's not advisable to be using manual system. Hostels without a management system are usually done manually.

Registration forms verification to other data saving processes are done manually and most at times, they are written on paper. Thus, a lot of repetitions can be avoided with an automated system. The drawbacks of existing systems lead to the design of a computerised system that will help reduce a lot of manual inputs. With this system in place, we can improve the efficiency of the system, thus overcome the drawbacks of the existing manual system.

This system is designed in favour of the hostel management which helps them to save the records of the students about their rooms and other things. It helps them from the manual work from which it is very difficult to find the record of the students and the mess bills of the students, and the information of about those ones who had left the hostel years before.

This system gives an idea about how a student and fee details, room allocation, mess expenditure is maintained in a better way. The hostel management system will also contain special features like how many students are in a room, student's id and free rooms or space available. The administration has a unique identity for each member as well as students' details.

Functional Requirements

STUDENT MODULE

This module is used to store student details i.e. information like profile details, contact information, educational details etc. Users can search according different criteria such as name, id, room number etc.

ROOM ALLOTMENT MODULE

This deals with allocation of room to students according to education details, section or course. Rooms will be allocated to students and an ID will be generated for it. It will display details students staying in the room or rooms. When a student leaves the room after the semester, the left date will be also saved.

ROOM FEES MODULE

This displays fee records, student dues status and balance amount status. It is also used to renew students rent every semester.

MESS BILL MODULE

This module keeps track of all transactions related to mess. The mess item expenditure for each student is calculated every month and mess bill for each student is calculated and displayed.

VISITORS MODULE

This allows the visitor details to view the visitors depending on various search criteria.

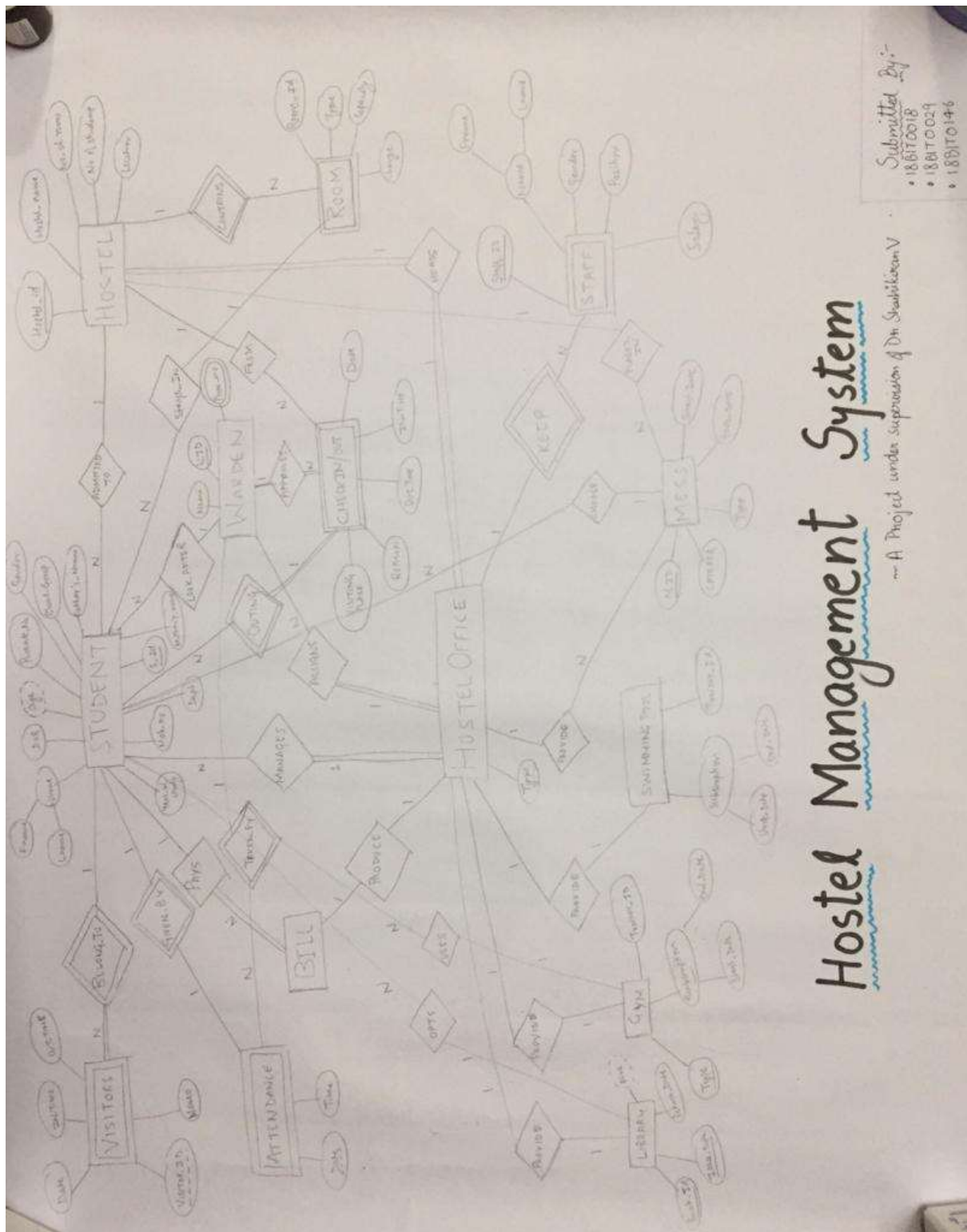
IN-OUT MODULE

This module keeps track of all the dates and time a student goes out of the hostel and where he goes to and also the return date for the same.

FACILITIES MODULE

This module keeps track of the subscriptions of the students to various facilities like swimming pool, library, gym, etc.

ER Diagram of the Hostel Management System



Relational Schema

Strong entities

1. HOSTEL OFFICE
2. LIBRARY
3. GYM
4. SWIMMING POOL
5. MESS
6. STUDENT
7. BILL
8. HOSTEL
9. WARDEN

Weak entities

1. VISITORS
2. ATTENDANCE
3. CHECK IN-OUT
4. ROOM
5. STAFF

Relationships

1. HOSTEL OFFICE->PROVIDES->LIBRARY
2. HOSTEL OFFICE->PROVIDES->GYM
3. HOSTEL OFFICE->PROVIDES->SWIMMING POOL
4. HOSTEL OFFICE->PROVIDES->MESS
5. HOSTEL OFFICE->KEEP->STAFF
6. STUDENT->CHOOSE->MESS
7. MESS->PLACED-IN->HOSTEL
8. VISITOTRS->BELONGS TO->STUDENT
9. ATTENDANCE->GIVEN BY->STUDENT
10. STUDENT->PAYS ->BILL
11. HOSTEL OFFICE->PRODUCE->BILL

12. HOSTEL OFFICE ->MANAGES -> STUDENT
13. STUDENT->OUTING->CHECK IN-OUT
14. STUDENT->ADMITTED TO ->HOSTEL
15. STUDENT->STAYS IN ->ROOM
16. ATTENDANCE->TAKEN BY->WARDEN
17. WARDEN->LOOK_AFTER->STUDENT
18. HOSTEL->CONTAINS->ROOM
19. HOSTEL OFFICE ->HEADS->HOSTEL
20. HOSTEL OFFICE ->ASSIGN->WARDEN
21. STUDENT->USES->GYM

MAPPING OF ER DIAGRAM TO RELATIONAL SCHEMA

HOW THE TABLES ARE IN BCNF

■ A relation schema R is in Boyce-Codd Normal Form (BCNF) if whenever an FD $X \rightarrow A$ holds in R, then X is a superkey of R

HOSTEL OFFICE

Attribute Name	Data Type	Constraint
Hos_type	Varchar2(30)	Primary Key
location	Varchar2(30)	Not Null
Contact_email	Varchar2(30)	Not Null

Constraints

- 1.Constraint Hosof_pk PRIMARY KEY(Hostype);
- 2.Constraint Loc_check CHECK(Location!=NULL)
3. Constraint Contact_check CHECK(Contact_email !=NULL)

Functional Dependencies

- 1.Hos_type->{Location,Contact_email};

<u>Hos_type</u>	Location	Contact_email
------------------------	-----------------	----------------------

HOSTEL

Attribute Name	Data Type	Constraint
Hostel_Id	Varchar2(30)	Primary Key
Hostel_name	Varchar2(30)	Not Null
No_of_rooms	Integer	Not Null
No_of_students	Integer	Not Null
Location	Varchar2(30)	Not Null

Constraints

- 1.Constraint Hos_pk PRIMARY KEY(Hos_type,Hostel_Id);
2. Constraint hos_fk3 FOREIGN KEY(Hos_type) references HOSTEL OFFICE(Hos_type);
3. Constraint hname_chek CHECK(Hostel_name!=NULL);
4. Constraint gen_chek CHECK(Gender!=NULL);
5. Constraint nro_chek CHECK(No_of_rooms!=NULL);
6. Constraint nstu_chek CHECK(No_of_students!=NULL);
7. Constraint loc_chek CHECK(Location!=NULL);

8.Constraint hos_fk2 FOREIGN KEY(W_Id) references WARDEN(W_Id);;

Functional Dependencies

1.Hos_type,Hostel_Id}-

>{Hostel_name,gender,No_of_rooms,No_of_students,Location,W_Id};

<u>Hos ty pe</u>	<u>Hostel Id</u>	Hostel_na me	No_of_roo ms	No_of_stude nts	Locatio n	W - Id
----------------------	----------------------	-----------------	-----------------	--------------------	--------------	--------------

ROOM

Attribute Name	Data Type	Constraint
Room Id	Integer	Primary Key
Type	Varchr2(30)	Not Null
Capacity	Number(30)	Not Null
Charge	Number(30)	Not Null

Functional Dependencies

1.{Hos_type,Hostel_Id,Room_Id}->{Type,Capacity};

2.{Type,Capacity}->{Charge};

ROOM DETAILS

Constraints

1.Constraint room_pak PARTIAL KEY(Room_Id);

2.Constraint room_pk PRIMARY KEY(Hos_type,Hostel_Id,Room_Id);

3.Constraint room_fk3 FOREIGN KEY(Hostel_Id) references
HOSTEL(Hostel_id);

4. Constraint room_fk3 FOREIGN KEY(Hos_type) references HOSTEL
OFFICE(Hos_type);

5. Constraint room_fk2 FOREIGN KEY(Capacity,Type) references
room_charges(Capacity,Type);

<u>Hos type</u>	<u>Hostel Id</u>	<u>Room Id</u>	Type	Capacity
-----------------	------------------	----------------	------	----------

ROOM CHARGES

Constraints

- 1.Constraint roomch_pk PRIMARY KEY(Type,Capacity);
2. Constraint roomch_chek2 CHECK(Charge!=NULL);

<u>Type</u>	<u>Capacity</u>	<u>Charge</u>
-------------	-----------------	---------------

STUDENT

Attribute Name	Data Type	Constraint
S_Id	Varchar2(30)	Primary Key
Fname	Varchar2(30)	Not Null
Lname	Varchar2(30)	Not Null
DOB	Date	Not Null
age	Number(2)	Derived
Gender	Varchar2(1)	Not Null
Parent_no	Number(10)	Not Null
Blood_group	Varchar2(2)	Not Null
Father_name	Varchar2(30)	Not Null
Mother_name	Varchar2(30)	Not Null
Dept	Varchar2(10)	Not Null
Year_of_study	Varchar2(5)	Not Null

Functional Dependencies

- 1.S_Id->
>{Fname,Lname,DOB,Gender,Parent_no,Blood_group,Father_name,Mother_name,Dept,Year_study,Hostel_Id};
- 2.Gender->{Hos_type};

Constraints

- 1.Constraint stu_pk PRIMARY KEY(S_Id);
2. Constraint stu_fk3 FOREIGN KEY(Hos_type) references HOSTEL OFFICE(Hos_type);

3. Constraint stu_fk3 FOREIGN KEY(Hostel_Id) references HOSTEL(Hostel_Id);
4. Constraint stu_chek1 CHECK(Fname!=NULL);
5. Constraint stu_chek2 CHECK(Lname!=NULL);
6. Constraint stu_chek3 CHECK(DOB!=NULL);
7. Constraint stu_chek4 CHECK(Gender!=NULL);
8. Constraint stu_chek5 CHECK(Parent_no!=NULL);
9. Constraint stu_chek6 CHECK(Blood_group!=NULL);
10. Constraint stu_chek7 CHECK(Father_name!=NULL);
11. Constraint stu_chek8 CHECK(Mother_name!=NULL);
12. Constraint stu_chek9 CHECK(Dept!=NULL);
13. Constraint stu_chek10 CHECK(Year_of_study!=NULL);

S_Id	Fname	Lname	DOB	Gender	Parent_no	Blood_group	Father_name	Mother_name	Dept	Year_of_study	Hostel_Id
------	-------	-------	-----	--------	-----------	-------------	-------------	-------------	------	---------------	-----------

Constraints

- 1.Constraint stuh_fk3 FOREIGN KEY(Hos_type) references HOSTEL OFFICE(Hos_type);
 2. Constraint stuh_fk3 FOREIGN KEY(Gender) references STUDENT(Gender);
 3. Constraint stuh_pk PRIMARY KEY(Gender);
- STUDENT HOSTEL TYPE

<u>Gender</u>	Hos_type
---------------	----------

WARDEN

ATTRIBUTE NAME	DATA TYPE	CONSTRAINT
W_Id	Varchar2(30)	Primary key
W_name	Varchar2(30)	Not null
Phone_no	Varchar2(30)	Multivalued
Email	Varchar2(30)	Not null

Constraints

- 1.Constraint war_pk PRIMARY KEY(W_Id);
- 2.Constraint war_chek CHECK(W_name!=NULL);
- 3.Constraint war_chek2 CHECK(Email !=NULL);

Functional Dependencies

- 1.W_Id->{W_name,Email,Phone_no};

<u>W_Id</u>	W_name	Email
--------------------	---------------	--------------

W_Id	Phone_no
-------------	-----------------

STAFF

ATTRIBUTE NAME	DATA TYPE	CONSTRAINT
Staff_Id	Varchar2(30)	Partial Key
Fname	Varchar2(30)	Not null
Lname	Varchar2(30)	Not null
Gender	Varchar2(6)	Not null
Position	Varchar2(30)	Not null

Salary	Number(10)	Not Null
---------------	------------	----------

Constraints

- 1.Constraint staff_fk2 FOREIGN KEY(Hos_type) references HOSTEL OFFICE(Hos_type);
- 2.Constraint staff_pk1 PRIMARY KEY(S_Id,Hos_type);
3. Constraint staff_chek CHECK(Fname !=NULL);
4. Constraint staff_chek2 CHECK(Lname !=NULL);
5. Constraint staff_chek3 CHECK(Gender!=NULL);
6. Constraint staff_chek4 CHECK(Position!=NULL);
7. Constraint staff_chek5 CHECK(Salary!=NULL);
- 8.Constraint staff_pak PARTIAL KEY(S_Id);

Functional Dependencies

- 1.{Hos_type,Staff_Id}->{Fname,Lname,Gender,Position,Salary};

<u>Hos_type</u>	<u>Staff_Id</u>	FName	Lname	Gender	Position	Salary
------------------------	------------------------	--------------	--------------	---------------	-----------------	---------------

BILL

ATTRIBUTE NAME	DATA TYPE	CONSTRAINT
Bill_Id	Varchar2(30)	Primary key
Amount	Number(15)	Not null
Reason	Varchar2(30)	Not null
Date	Date	Not null
Payment_Date	Date	Any from domain(can be null)

Constraints

- 1.Constraint bill_pk PRIMARY KEY(Bill_Id);

2. Constraint bill_chek1 CHECK(Amount!=NULL);
3. Constraint bill_chek2 CHECK(Reason!=NULL);
4. Constraint bill_chek3 CHECK(Date!=NULL);
5. Constraint bill_fk2 FOREIGN KEY(S_Id) references STUDENT(S_Id);

Functional Dependencies

1. Bill_id → {S_Id, Amount, Reason, Date, Payment_Date};

<u>Bill_Id</u>	S_Id	Amount	Reason	Date	Payment_Date
----------------	------	--------	--------	------	--------------

ATTENDANCE

ATTRIBUTE NAME	DATA TYPE	CONSTRAINT
Date	Date	Partial key
Marked	Varchar2(1)	Not null
Time	Datetime	Not null

Constraints

1. Constraint att_fk FOREIGN KEY(S_Id) references STUDENT(S_Id);
2. Constraint att_pk PRIMARY KEY(S_Id, Date);
3. Constraint att_fk PARTIAL KEY(Date);
4. Constraint att_chek1 CHECK(Marked!=NULL);
5. Constraint att_chek3 CHECK(Time!=NULL);

Functional Dependencies

1. {S_Id, Date} → {Marked, Time};

<u>S_Id</u>	Date	Marked	Time
-------------	------	--------	------

CHECK IN/OUT

ATTRIBUTE NAME	DATA TYPE	CONSTRAINT
Date	Date	Partial key
Reason	date	Not null
Visiting_Place	Varchar2(20)	Not null
Out_time	Datetime	Not null
In_time	Datetime	Any from domain

Constraints

- 1.Constraint ch_fk3 FOREIGN KEY(S_Id) references STUDENT(S_Id);
- 2.Constraint ch_pak PARTIAL KEY(Date);
- 3.Constraint ch_pk PRIMARY KEY(S_Id,Date);
- 4.Constraint ch_chek3 CHECK(Reason!=NULL);
- 5.Constraint ch_chek2 CHECK(Out_time!=NULL);
- 6.Constraint ch_chek1 CHECK(Visiting_Place!=NULL);

Functional Dependencies

- 1.{S_Id,Date}->{Reason,Out_time,Visiting_Place,In_time};

<u>S_Id</u>	<u>Date</u>	Reason	Visiting_Place	Out_time	In_time
--------------------	--------------------	---------------	-----------------------	-----------------	----------------

MESS

ATTRIBUTE NAME	DATA TYPE	CONSTRAINT
<u>M_Id</u>	Varchar2(30)	Primary Key
Caterer	Varchar2(30)	Not Null
Type	Varchar2(10)	Not null
Start_Date	Date	Not null
End_Date	Date	Not null

Charge	Number(10)	Not null
---------------	------------	----------

Functional Dependencies

1.M_Id,Hos_type,Hostel_Id}->{Type,Caterer,Charge};

2.S_Id->{M_Id,Start_Date,End_Date};

Constraints

1.Constraint mess_fk3 FOREIGN KEY(Hos_type) references HOSTEL OFFICE(Hos_type);

2.Constraint mess_fk2 FOREIGN KEY(Hostel_Id) references HOSTEL(Hostel_Id);

3.Constraint mess_pk PRIMARY KEY(M_Id,Hos_type,Hostel_Id);

4.Constraint mess_chek3 CHECK(Caterer!=NULL);

5.Constraint mess_chek2 CHECK(Type!=NULL);

6.Constraint mess_chek1 CHECK(Charge!=NULL);

<u>M_Id</u>	<u>Hos_type</u>	<u>Hostel_Id</u>	Caterer	Type	Charge
--------------------	------------------------	-------------------------	----------------	-------------	---------------

STUDENT MESS

Constraints

1.Constraint smess_fk3 FOREIGN KEY(S_Id) references STUDENT(S_Id);

2.Constraint smess_fk2 FOREIGN KEY(M_Id) references Mess(M_Id);

3.Constraint smess_pk PRIMARY KEY(S_Id);

4.Constraint smess_chek3 CHECK(Start_Date!=NULL);

5.Constraint smess_chek2 CHECK(End_Date!=NULL);

<u>S_Id</u>	M_Id	Start_Date	End_Date
--------------------	-------------	-------------------	-----------------

SWIMMING POOL

ATTRIBUTE NAME	DATA TYPE	CONSTRAINT
<u>Start Date</u>	Date	Not null
End_Date	Date	Not Null
Trainer_Id	Varchar2(10)	Not null
Pool_Type	Varchar2(10)	Primary Key
Charge	Number(10)	Not null

Functional Dependencies

1. {Pool_Type, Hos_type} -> {Trainer_Id, Charge};
2. S_Id -> {Pool_Type, Start_Date, End_Date};

Constraints

1. Constraint pool_fk3 FOREIGN KEY(Hos_type) references HOSTEL OFFICE(Hos_type);
2. Constraint pool_pk PRIMARY KEY {Pool_Type, Hos_type};
3. Constraint pool_chek3 CHECK(Trainer_Id!=NULL);
4. Constraint pool_chek1 CHECK(Charge!=NULL);

<u>Pool_Type</u>	<u>Hos_type</u>	Trainer_Id	Charge
------------------	-----------------	------------	--------

STUDENT POOL

Constraints

1. Constraint spool_fk3 FOREIGN KEY(S_Id) references STUDENT(S_Id);
2. Constraint spool_fk2 FOREIGN KEY(Pool_Type) references SWIMMING POOL(Pool_Type);

- 3.Constraint spool_pk PRIMARY KEY(S_Id);
- 4.Constraint spool_chek3 CHECK(Start_Date!=NULL);
- 5.Constraint spool_chek2 CHECK(End_Date!=NULL);

<u>S_Id</u>	Pool_Type	Start_Date	End_Date
--------------------	------------------	-------------------	-----------------

GYM

ATTRIBUTE NAME	DATA TYPE	CONSTRAINT
<u>Start_Date</u>	Date	Not null
<u>End_Date</u>	Date	Not Null
<u>Trainer_Id</u>	Varchar2(10)	Not null
<u>Gym_Type</u>	Varchar2(10)	Primary Key
<u>Charge</u>	Number(10)	Not null

Functional Dependencies

- 1.{Gym_Type,Hos_type}->{Trainer_Id,Charge};
- 2.S_Id->{Gym_Type,Start_Date,End_Date};

Constraints

- 1.Constraint gym_fk3 FOREIGN KEY(Hos_type) references HOSTEL OFFICE(Hos_type);
- 2.Constraint gym_pk PRIMARY KEY{Gym_Type,Hos_type};
- 3.Constraint gym_chek3 CHECK(Trainer_Id!=NULL);
- 4.Constraint gym_chek1 CHECK(Charge!=NULL);

<u>Gym_Type</u>	<u>Hos_type</u>	Trainer_Id	Charge
------------------------	------------------------	-------------------	---------------

STUDENT GYM

Constraints

- 1.Constraint sgym_fk3 FOREIGN KEY(S_Id) references STUDENT(S_Id);
- 2.Constraint sgym_fk2 FOREIGN KEY(Gym_Type) references SWIMMING POOL(Pool_Type);
- 3.Constraint sgym_pk PRIMARY KEY(S_Id);
- 4.Constraint sgym_chek3 CHECK(Start_Date!=NULL);
- 5.Constraint sgym_chek2 CHECK(End_Date!=NULL);

<u>S_Id</u>	Gym_Type	Start_Date	End_Date
--------------------	-----------------	-------------------	-----------------

LIBRARY

ATTRIBUTE NAME	DATA TYPE	CONSTRAINT
<u>Lib_Id</u>	Date	Primary Key
Total_Books	Date	Not Null
Book_name	Varchar2(10)	Not null
Author	Varchar2(10)	Not null
Quantity	Number(10)	Not null
Manager	Varchar2(15)	Not null
Issue_Date	Date	Not null
Return_Date	Date	Not null
Fine	Number(10)	Derived

Functional Dependencies

- ▶ {Lib_Id,Hos_type}->Manager,Total_Books,Book_name};
- ▶ {Book_name,Lib_Id}>{Quantity};
- ▶ Book_name->Author
- ▶ S_Id->Book_name;
- ▶ {S_Id,Book_name,Issue_Date}->{Return_Date,Fine}

Constraints

- ▶ Constraint lib_fk3 FOREIGN KEY(Hos_type) references HOSTEL OFFICE(Hos_type);
- ▶ Constraint lib_pk PRIMARY KEY(Lib_Id);
- ▶ Constraint lib_chek3 CHECK(Manager!=NULL);
- ▶ Constraint lib_chek1 CHECK(Total_Books!=NULL);

<u>Lib_Id</u>	<u>Hos_type</u>	Manager	Total_Books
---------------	-----------------	---------	-------------

BOOKS

Constraints

- ▶ Constraint book_fk3 FOREIGN KEY(Book_name) references LIBRARY(Book_name);
- ▶ Constraint book_fk2 FOREIGN KEY(Lib_Id) references LIBRARY(Lib_Id);
- ▶ Constraint book_pk PRIMARY KEY(Book_name,Lib_Id);
- ▶ Constraint book_chek3 CHECK(Author!=NULL);
- ▶ Constraint book_chek2 CHECK(Quantity!=NULL);

<u>Book_name</u>	<u>Lib_Id</u>	Author	Quantity
------------------	---------------	--------	----------

STUDENT ISSUE

Constraints

- ▶ Constraint sbook_fk3 FOREIGN KEY(S_Id) references STUDENT(S_Id);
- ▶ Constraint sbook_fk2 FOREIGN KEY(Book_name) references BOOKS(Book_name);
- ▶ Constraint sbook_pk PRIMARY KEY(S_Id,Book_name,Issue_Date);
- ▶ Constraint sbook_chek2 CHECK(Return_Date!=NULL);

<u>S Id</u>	<u>Book name</u>	<u>Issue Date</u>	<u>Return_Date</u>
-------------	------------------	-------------------	--------------------

RELATION EXTENSION

HOSTEL OFFICE

<u>Hos_type</u>	<u>Location</u>	<u>Contact_email</u>
Mens	Near SJT	hostel.1@gmail.com
Ladies	Near TT	hostel.2@gmail.com

HOSTEL

<u>Hos_type</u>	<u>Hostel Id</u>	<u>Hostel_name</u>	<u>No_of_rooms</u>	<u>No_of_student</u>	<u>Location</u>	<u>W_Id</u>
Mens	A	A block	200	900	Near SJT	A-001
Mens	B	B block	180	240	Near TT	B-002
Mens	C	C block	190	300	Near SMV	C-003
Ladies	A	D block	200	600	Near Library	D-004
Ladies	B	E block	190	300	Near Main Gate	E-005

ROOM

<u>Hos_type</u>	<u>Hostel Id</u>	<u>Room Id</u>	<u>Type</u>	<u>Capacity</u>
Mens	A	100	AC	6
Mens	B	101	NAC	4
Mens	C	102	NAC	5

Ladies	A	103	AC	3
Ladies	B	104	AC	2

ROOM CHARGES

<u>Type</u>	<u>Capacity</u>	<u>Charge</u>
AC	6	50000
NAC	4	60000
NAC	5	30000
AC	3	75000
AC	1	100000

STUDENT

S_Id	Fname	Lname	DOB	Gender	Parent_no	Blood_Group	Father_name	Mother_name	Dept	Year_of_study	Hostel_Id
18BIT0001	Ami	Cat	22-08-96	F	9456212199	O+	Ram Singh	Lal Devi	IT	4	A
18BIT0002	Anu	Pia	22-03-97	F	9818097731	A+	Shyam Sethi	Peei Devi	IT	4	B
18BCL0018	Dev	Klo	02-08-99	M	9567685166	O+	Amit Mishra	Shashi Devi	CE	4	A
18BME0029	Raj	Sia	23-07-98	M	9568763057	B+	Sudhir Sia	Mia Devi	ME	4	C
18BEE0146	Jon	Sno	13-04-99	M	9714176526	A-	Ram Sno	Ema Watson	EE	4	B

STUDENT HOSTELTYPE

<u>Gender</u>	<u>Hos_type</u>
Male	Men
Female	Ladies

WARDEN

<u>W_Id</u>	<u>W_name</u>	<u>Email</u>
-------------	---------------	--------------

A-001	Praneeth	praneeth@gmail.com
B-002	Kartikey	kartikey@gmail.com
C-003	Piyush	piyush@gmail.com
D-004	Awadh	awadh@gmail.com
E-005	Rishab	rishab@gmail.com

WARDEN CONTACT

W_Id	Phone_no
A-001	9897600543
A-001	9986758541
C-003	9896745643
C-003	8791309225
E-005	8755277277

STAFF

Hos_type	Staff_Id	Fname	Lname	Gender	Position	Salary
Mens	S-100	Ravi	Shankar	Male	Clerk	200000
Mens	S-101	Brisilla	M reddy	Female	Matron	150000
Ladies	S-102	Chandra	Deva	Male	Clerk	300000
Ladies	S-103	Ravi	Goswami	Male	Computer operator	400000
Mens	S-104	Aditya	Shukla	Male	Supervisor	95000

BILL

Bill_Id	S_Id	Amount	Reason	Date	Payment_date
A-10000	18BIT0018	100000	Fees	2-10-2019	09-11-2019
A-10001	18BIT0029	30000	Mess	5-10-2019	19-12-2019

A-10002	18BIT0146	450000	Hostel	4-10-2019	09-11-2019
A-10003	18BIT0035	550000	Security	5-10-2019	09-11-2019
A-10004	18BIT0019	450000	Gym	4-09-2019	06-11-2019

ATTENDANCE

<u>S_Id</u>	<u>Date</u>	<u>Marked</u>	<u>Time</u>
18BIT0018	2-10-2019	A	10:00 AM
18BIT0029	2-10-2019	P	10:00 AM
18BIT0146	4-10-2019	P	08:00 AM
18BIT0035	3-10-2019	A	10:00 AM
18BIT0019	5-10-2019	P	08:00 AM

CHECK IN/OUT

<u>S_Id</u>	<u>Date</u>	<u>Reason</u>	<u>Visiting_Place</u>	<u>Out_time</u>	<u>In_time</u>
18BIT0018	3-10-2019	Meeting	Chennai	10:00 AM	01:00 AM
18BIT0029	5-10-2019	Movie	Vellore	11:00 PM	02:00 PM
18BIT0146	8-10-2019	Work	Bangalore	01:00 AM	08:00 AM
18BIT0035	16-10-2019	Movie	Bangalore	05:00 PM	9:00 PM
18BIT0019	19-10-2019	Meeting	Chennai	09:00 PM	07:00 AM

MESS

<u>M_Id</u>	<u>Hostel type</u>	<u>Hostel Id</u>	<u>Caterer</u>	<u>Type</u>	<u>Charge</u>
M-001	Mens	A	Darling	Veg	25000
M-002	Mens	B	Zenith	Non-veg	28000
M-003	Mens	C	PR	Special	30000
M-004	Ladies	A	Darling	Non-veg	28000
M-005	Ladies	B	Zenith	Veg	25000

STUDENT MESS

<u>S_Id</u>	<u>M_Id</u>	<u>Start_Date</u>	<u>End_Date</u>
18BIT0018	M-003	02-10-2019	02-10-2020
18BIT0029	M-003	06-10-2019	6-10-2020
18BIT0146	M-001	18-11-2019	18-11-2020
18BIT0035	M-002	07-11-2019	07-11-2020
18BIT0019	M-005	09-11-2019	09-11-2020

SWIMMING POOL

<u>Pool_Type</u>	<u>Hostel_type</u>	<u>Trainer_Id</u>	<u>Charge</u>
Fibre glass	Mens	T-100	10000
Vinyl liner	Mens	T-101	12000
Concrete	Mens	T-102	15000
Fibre glass	Ladies	T-103	10000
Concrete	Ladies	T-104	12000
Vinyl liner	Ladies	T-105	15000

STUDENT POOL

<u>S_id</u>	<u>Pool_Type</u>	<u>Start_Date</u>	<u>End_Date</u>
18BIT0029	Vinyl lyner	2-10-2019	2-10-2020
18BIT0018	Fibre glass	3-11-2019	3-10-2020
18BIT0035	Fibre glass	8-10-2019	8-10-2020
18BIT0019	Concrete	20-10-2019	20-10-2020

GYM

<u>Gym_Type</u>	<u>Hostel_type</u>	<u>Trainer_Id</u>	<u>Charge</u>
AC	Mens	T-200	20000
NAC	Mens	T-201	15000

AC	Ladies	T-202	20000
NAC	Ladies	T-203	15000

STUDENT GYM

<u>S_Id</u>	Gym_Type	Start_Date	End_Date
18BIT0146	AC	2-10-2019	2-10-2020
18BIT0029	AC	3-10-2019	3-10-2020
18BIT0018	NAC	8-10-2019	8-10-2019

LIBRARY

<u>Lib_Id</u>	<u>Hos_type</u>	Manager	Total_Books
L-100	Mens	Kartika	40000
L-101	Ladies	Vaishali	50000

BOOKS

<u>Book_name</u>	<u>Lib_Id</u>	Author	Quantity
DBMS	L-100	Ravi prashad	10
Web tech	L-101	Ojasvi	12
DSA	L-100	Devang	45
Calculus	L-101	Priyanshu	23

STUDENT ISSUE

<u>S_Id</u>	Book_name	Issue_Date	Return_Date
18BIT0029	DBMS	2-10-2019	30-10-2019
18BIT0018	DSA	5-10-2019	2-11-2019
18BIT0146	DBMS	2-10-2019	3-11-2019
18BIT0035	Calculus	8-10-2019	17-10-2019

SQL

```
1 create table hosteloffice(  
2   hos_type varchar2(30) constraint ho_pk primary key,  
3   location varchar2(30) constraint loc_check not null,  
4*  contact_email varchar2(30) constraint contact_check not null)  
SQL>  
SQL> /
```

Table created.

```
SQL> create table hostelid(hostel_id varchar2(30),  
2   constraint hos_pk primary key(hostel_id))  
3 /
```

Table created.

```
1 create table hostel(  
2   hostel_id varchar2(30),  
3   hostel_name varchar2(30) not null,  
4   no_of_rooms integer not null,  
5   no_of_students integer not null,  
6   location varchar2(30) not null,  
7   hos_type varchar2(30),  
8   constraint hos_fk1 foreign key(hostel_id) references hostelid(hostel_id),  
9   constraint hos_fk foreign key(hos_type) references hosteloffice(hos_type),  
10*  constraint host_pk primary key(hos_type,hostel_id))  
SQL> /
```

Table created.

```
1 create table room_charges (  
2   type varchar2(30),  
3   capacity number(30),  
4   charge number(30) not null,  
5*  constraint rc_pk primary key(type,capacity))  
SQL> /
```

Table created.

```

1 create table room (
2     room_id integer ,
3     type varchar2(30) ,
4     capacity number(30) ,
5     hos_type varchar2(30),
6     hostel_id varchar2(30),
7     constraint room_fk2 foreign key(type,capacity) references room_charges(type,capacity),
8     constraint room_fk1 foreign key(hos_type) references hosteloffice(hos_type),
9     constraint room_fk3 foreign key(hostel_id) references hostelid(hostel_id),
10* constraint room_pk primary key(hos_type,hostel_id,room_id))
SQL> /

Table created.

SQL>

```

```

SQL> create table student_hosteltype(
2     gender char(1),
3     hos_type varchar2(30),
4     constraint sht_pk primary key(gender),
5     constraint sht_fk foreign key(hos_type) references hosteloffice(hos_type));

Table created.

```

```

1 create table student (
2     s_id varchar2(30),
3     fname varchar2(30) not null,
4     lname varchar2(30) not null,
5     dob date not null,
6     gender char(1) not null,
7     parent_no number(10) not null,
8     blood_group varchar2(2) not null,
9     father_name varchar2(30) not null,
10    mother_name varchar2(30) not null,
11    dept varchar2(10) not null,
12    year_of_study varchar2(5) not null,
13    hostel_id varchar2(30),
14    constraint stu_pk primary key(s_id),
15* constraint stu_fk foreign key(hostel_id) references hostelid(hostel_id))
SQL> /

Table created.

SQL>

```

```

SQL> create table warden(
2     w_id varchar2(30),
3     w_name varchar2(30) not null,
4     email varchar2(30) not null,
5     constraint w_pk primary key(w_id));

Table created.

SQL> create table warden_pn(
2     w_id varchar2(30),
3     phone_no varchar2(30),
4     constraint wp_fk foreign key(w_id) references warden(w_id));

Table created.

```

```

1 create table staff(
2   hos_type varchar2(30),
3   staff_id varchar2(30),
4   fname varchar2(30) not null,
5   lname varchar2(30) not null,
6   gender varchar2(6) not null,
7   position varchar2(30) not null,
8   salary number(10) not null,
9   constraint staff_pk primary key(staff_id),
10* constraint staff_fk foreign key(hos_type) references hostelloffice(hos_type))
SQL> /

```

Table created.

```

SQL>
SQL> create table bill(
2   bill_id varchar2(30),
3   s_id varchar2(30),
4   amount number(15) not null,
5   reason varchar2(30) not null,
6   dateof date not null,
7   payment_date date,
8   constraint bill_pk primary key(bill_id),
9   constraint bill_fk foreign key(s_id) references student(s_id));

```

Table created.

```

1 create table attendance (
2   s_id varchar2(30),
3   dateof date ,
4   marked varchar2(1) not null,
5   timeof timestamp not null,
6   constraint att_fk foreign key(s_id) references student(s_id),
7* constraint att_pk primary key(s_id,dateof))
SQL> /

```

Table created.

```

1 create table checkin_out (
2   s_id varchar2(30),
3   dateof date ,
4   reason varchar2(30) not null,
5   visiting_place varchar2(20) not null,
6   out_time timestamp not null,
7   in_time timestamp,
8   constraint ch_fk foreign key(s_id) references student(s_id),
9* constraint ch_pk primary key(s_id,dateof))
SQL> /

```

Table created.

```

SQL> create table messid(m_id varchar2(30),
2   constraint mid_pk primary key(m_id))
3   /

```

Table created.

```

1 create table mess(
2 m_id varchar2(30),
3 hos_type varchar2(30),
4 hostel_id varchar2(30),
5 caterer varchar2(30),
6 type varchar2(30),
7 charge number(10),
8 constraint mess_fk1 foreign key(hos_type) references hostelloffice(hos_type),
9 constraint mess_fk2 foreign key(hostel_id) references hostelid(hostel_id),
10* constraint mess_pk primary key(hos_type,hostel_id,m_id))
SQL> /

```

Table created.

```

SQL> create table student_mess(
2 s_id varchar2(30),
3 m_id varchar2(30),
4 start_date date,
5 end_date date,
6 constraint st_mess_fk1 foreign key(s_id) references student(s_id),
7 constraint st_mess_fk2 foreign key(m_id) references messid(m_id),
8 constraint st_mess_pk primary key(s_id))
9 /

```

Table created.

SQL>

```

1 create table pooltype(
2 pool_type varchar2(30),
3* constraint pt_pk primary key(pool_type))
SQL> /

```

Table created.

```

1 create table swimming_pool(
2 pool_type varchar2(30),
3 hos_type varchar2(30),
4 trainer_id varchar2(10),
5 charge number(10) not null,
6 constraint sp_fk foreign key(pool_type) references pooltype(pool_type),
7 constraint sp_fk2 foreign key(hos_type) references hostelloffice(hos_type),
8* constraint sp_pk primary key(pool_type,hos_type))
QL> /

```

able created.

```

SQL> create table student_pool(
2 s_id varchar2(30),
3 pool_type varchar2(30),
4 start_date date not null,
5 end_date date not null,
6 constraint stp_fk foreign key(s_id) references student(s_id),
7 constraint stp_fk2 foreign key(pool_type) references pooltype(pool_type),
8 constraint stp_pk primary key(s_id));

```

Table created.

```

QL> create table gymtype(
2  gym_type varchar2(10),
3  constraint gt_pk primary key(gym_type));

able created.

QL> create table gym(
2  gym_type varchar2(10),
3  hos_type varchar2(30),
4  trainer_id varchar2(10) not null,
5  charge number(10) not null,
6  constraint g_fk foreign key(gym_type) references gymtype(gym_type),
7  constraint g_fk2 foreign key(hos_type) references hostelloffice(hos_type),
8  constraint g_pk primary key(gym_type,hos_type));

able created.

QL> create table student_gym(
2  s_id varchar2(30),
3  gym_type varchar2(10),
4  start_date date not null,
5  end_date date not null,
6  constraint st_fk foreign key(s_id) references student(s_id),
7  constraint st_fk2 foreign key(gym_type) references gymtype(gym_type),
8  constraint st_pk primary key(s_id));

able created.

```

```

C:\WINDOWS\system32\cmd.exe - sqlplus

1 create table libid(
2  lib_id varchar2(10),
3* constraint li_pk primary key(lib_id))
SQL> /

Table created.

SQL> create table library(
2  lib_id varchar2(10),
3  hos_type varchar2(30),
4  manager varchar2(15) not null,
5  total_books number(10) not null,
6  constraint l_fk foreign key(lib_id) references libid(lib_id),
7  constraint l_fk2 foreign key(hos_type) references hostelloffice(hos_type),
8  constraint l_pk primary key(lib_id));

Table created.

```

```

1 create table books(
2  book_name varchar2(30),
3  lib_id varchar2(10),
4  author varchar2(20) not null,
5  quantity number(10) not null,
6  constraint bk_fk foreign key(lib_id) references libid(lib_id),
7* constraint bk_pk primary key(book_name,lib_id))
SQL> /

Table created.

```



```

SQL> create table student_issue(
  2  s_id varchar2(30),
  3  book_name varchar2(30),
  4  issue_date date not null,
  5  return_date date,
  6  constraint si_fk foreign key(s_id) references student(s_id),
  7  constraint si_fk2 foreign key(book_name) references books(book_name),
  8  constraint si_pk primary key(s_id,book_name,issue_date));
Table created.

```

TRIGGERS

1. To store the details of room

```

SQL> create table roomhistory(room_id integer,type varchar2(30),capacity number(30),hos_type varchar2(30),hostel_id varchar2(30))
  2  /
Table created.

SQL> CREATE OR REPLACE TRIGGER DeleteRoom
  2  BEFORE DELETE ON room
  3  FOR EACH ROW
  4  BEGIN
  5  INSERT INTO roomhistory
  6  VALUES(:Old.room_id,:Old.type,:Old.capacity,:Old.hos_type,:Old.hostel_id);
  7  END DeleteRoom;
  8  /
Trigger created.

```

```
SQL> delete from room where room_id='100';
```

1 row deleted.

```
SQL> select * from roomhistory;
```

ROOM_ID	TYPE	CAPACITY	HOS_TYPE	HOSTEL_ID
100	AC	6	A	

2. To store the details of bill

```

1 CREATE OR REPLACE TRIGGER update_bill
2 AFTER UPDATE
3 ON bill FOR EACH ROW
4 BEGIN
5 INSERT INTO bill_audit
6 VALUES
7 (:old.bill_id,:old.s_id,:old.amount,:old.dateof,:old.payment_date);
8* END update_bill;
SQL> /

```

STORED PROCEDURES

A Procedure is a subprogram unit that consists of a group of PL/SQL statements.

Each procedure in Oracle has its own unique name by which it can be referred. This

subprogram unit is stored as a database object.

- Procedures are standalone blocks of a program that can be stored in the database.
- Call to these procedures can be made by referring to their name, to execute the PL/SQL statements.
- It is mainly used to execute a process in PL/SQL.
- It can have nested blocks, or it can be defined and nested inside the other blocks or packages.
- It contains declaration part (optional), execution part, exception handling part (optional).
- The values can be passed into the procedure or fetched from the procedure through parameters.
- These parameters should be included in the calling statement.
- Procedure can have a RETURN statement to return the control to the calling block, but it cannot return any values through the RETURN statement.
- Procedures cannot be called directly from SELECT statements. They can be called from another block or through EXEC keyword.

1. Retrieve student id and parent number of students having more than 3 years of study.

```
SQL>create or replace procedure stud_det3
```

```
is
```

```
sid student.s_id%type;
```

```
number student.parent_no%type;
```

```
begin
```

```

loop
select s_id into sid, parent_no into number from student where
year_of_study>3;
dbms_output.put_line(sid||" "||number);
end loop;
end;
/

```

2. Retrieve student id and bill amount of students who have not submitted their generated bills

```

SQL>create or replace procedure bill_retr
is
sid bill.s_id%type;
bamount bill.amount%type;
begin
loop
select s_id into sid,amount into bamount from bill where
payment_date=NULL;
dbms_output.put_line(sid||" "||bamount);
end loop;
end;
/

```