CHAPTER 1

INTRODUCTION

The Project HOTEL DATABASE is a web-based project cum system that is specially designed for a tracking various sector of hotels, which gives information related to the customers, rooms, staffs and other various peoples. This project develops a system that can be used by the hotel management to keep track of all the details related to reservations history, diseases and other various information can be recorded using this system. It automates the communication between all of them who are related to hotel management.

Moreover, this system is designed for the particular need of the hotel to carry out their operations in a smooth and effective manner. This project has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and, in some cases, reduce the hardships faced by the existing system.

The application is used to reduce as much as errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is required for the user to use this system. Thus, it proves to be user friendly. Hotel Database, as described above can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather than concentrating on record keeping. Thus, it will help organization in better utilization of resources and it will reduce their time to update the data in the systems compared to manually updating data.

Every organization whether it is big or small has many challenges to overcome and managing those challenges is very important by using this system all those challenges can be overcome and the information of customers, employees, rooms, reservations, and other information related to it can be easily managed using this Hotel Database. Every Hotel has different needs according to their use of the system therefore we design exclusive Database management systems that are adaptable to their requirements.

This system is designed to assist in planning and maintain the records whenever required in case of some emergency and will help us ensure that our organization is equipped with the right level of information and details for our future goals. Also, for those busy customers who are always on the go, our systems come with all remote access features which will allow us to manage our workforce anytime. These systems will ultimately allow us to better manage our resources.

Before they were using the paper forms, ledgers and old records for monitoring information related to reservation of customers, discharge of patients and payment details of customers and salary information of employees and other working staffs. After using this system, we reduce all the paper work and all those works can be done using the automated system of hotel database. Now a days there are many records generated in today's world and it is very useful that all those data has been stored in correct manner and efficient manner, in order to store all those data, we have developed an automated system which can be useful in storing those various records and data and this system is very secure and reliable.

This system is designed to assist in planning and maintain the records whenever required in case of some emergency and will help us ensure that our organization is equipped with the right level of information and details for our future goals. Also, for those customer who are always on the go, our systems come with all remote access features which will allow us to manage our workforce anytime. These systems will ultimately allow us to better manage our resources

The functionalities provided by the present Hotel Database include:

- Provides the searching facilities based on various factors such as customers, employees, department, salary, Room's Availability and etc.,
- It manages the reservation details for every customer and also keeps payment details.
- Manages the information of the rooms.
- Shows the details of the customers, departments and reservations.
- Keeps track of the available rooms.
- Supports any number of customers addition.
- Can support the access of multiple customers at one instant.
- All the update and delete operations are recorded for audit purposes.

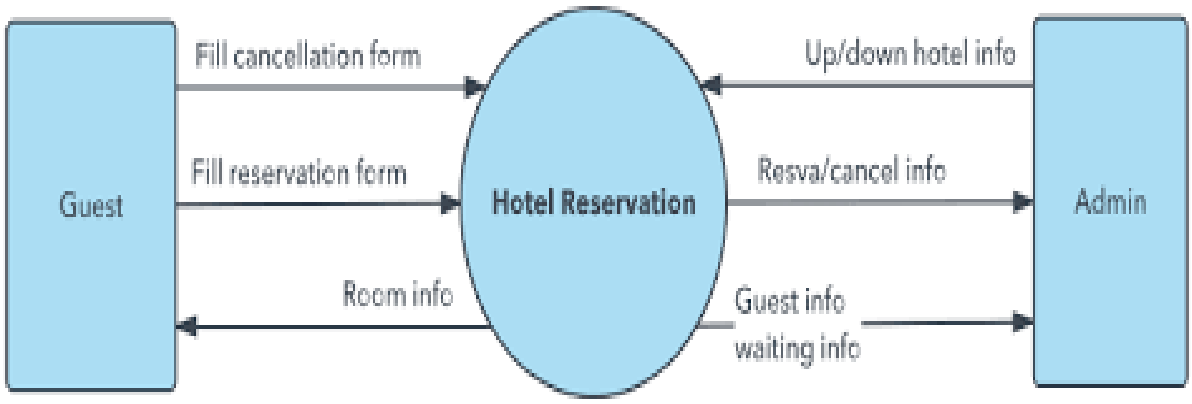

Figure 1.1 Pictorial view of Hotel Database

CHAPTER 2

PROBLEM STATEMENT

A small hotel needs a system to manage the database of the hotel. They have an extremely limited budget, and do not want to purchase an expensive software suite in such a risky economic environment. In a single software package, they will need the ability to handle all patients' entry and exit record, the details of their departments, salary of employees and other staff members and also to maintain an efficient way to store all these data.

Purpose: To create an affordable Hotel Database for a small Hotel as an alternative to similar high dollar systems on the market today. To stay competitive, an online website will also be implemented to reach a wider section of customers to get hotel and to get well infrastructure and also to post reviews of various customers with various experience and to keep track of other details of hotel in order to keep hotel running smoothly. The system will be implemented using HTML, PHP, CSS and MySQL..

2.1 OBJECTIVES

The Main Objective of the Hotel Database is to provide a well maintained and well efficient Hotel Database Management System. Hotels today currently use manual systems for managing and maintaining the information of their Database. So, the Objective here is to remove current systems use of numerous paper forms that stores data and spread throughout the hotel management infrastructure. Information on paper are always incomplete and it does not give proper security of those records and does not follow management standards. Paper forms are always lost in case of some emergency and in case of a management's audit process. Multiple Copies of that information may be available in some records and it may

lead to inconsistency of that record and it may be difficult to identify the original one so, here the objective is to eliminate all those paper forms and bring a well maintained and efficient Database that reduces all the hectic work of papers with a management systemand with the help of this system the need of maintaining records on large books and various methods are reduced. They have an extremely limited budget, and do not want to purchase an expensive software suite in such a risky economic environment. In a single software package, they will need the ability to handle all patients' entry and exit record, the details of their departments, salary of employees and other staff members and also to maintain an efficient way to store all these data.

So,the Objective here is to remove current systems use of numerous paper forms that stores data and spread throughout the hospital management infrastructure. Information on paper are always incomplete and it does not give proper security of those records and does not follow management standards. Paper forms are always lost in case of some emergency and in case of a management's audit process. Multiple Copies of that information may be available in some records and it may lead to inconsistency of that record and it may be difficult to identify the original one so, here the objective is to eliminate all those paper forms and bring a well maintained and efficient Database that reduces all the hectic work of papers with a management system and with the help of this system the need of maintaining records on large books and various methods are reduced.

Hotel Database system we can easily reduce the manual work and can generate more securable and more efficient database which can record various data related to hotel infrastructure and also by using this system man work is also reduced by that was previously

done using paper and forms. This project has more scope in the future and can be integrated further. This project is successfully implemented with all the features mentioned earlier. Deployment of this application will help the hotel management to reduce the unnecessary wastage of time in doing manually. Therefore, we are successfully able to reach the goals and target of the project. Thus, keeping the working of the manual system as the basis of our system. We have developed an automated version of the manual system, named as Hotel Database. This project Hotel Databases aimed to develop and maintain the day-to-day state of admission, discharge, fee paid of patients, list of doctors, reports generation, and do many more works automatically

2.2 DATABASE ASSUMPTIONS

A Hotel Database basically has many important elements such as customers, rooms, payments, departments, salary's and etc., Each element must be tracked based on its availability. Each element must be related to one another in order to achieve proper relation between them. The Assumptions made on Hotel Database is as follows...

- Customer have an id, first name, last name, phone no and address
- Room have an room no, room type, no of beds, mobile no.
- Each customer may have zero or many rooms reserved who have customer id, room no, check in date and check out date as attributes in booking table
- Department has an id, name, location, phone no
- Employee has id, name, work, salary, ph no, address
- Each department may have one or more employee
- Payment has transaction id, payment mode, payment date and total amount.

- Every payment id is associated to one customer.
- Every customer has to pay amount based on the time type of room and requirements
- Each department has one or more employee associated to its department

CHAPTER 3

FEASIBILITY STUDY

Once the system objectives have been ascertained by initial investigation, we need to spell the various possible solutions to meet the various objectives. The feasibility study is conducted to check whether the candidate system is feasible. The system which is selected to be the best against the criteria is thereafter designed and developed. The feasibility study takes into consideration. The risks involved in the project development beforehand. Feasibility study includes seven distinct but inters related type of feasibility. All these feasibility study used are used. Focus is on establishing whether the technology needed for the proposed system is available and how this technology can be integrated within the organization. Technologies include are:

- Hardware
- Software
- Application developed environment

3.1 ECONOMIC FEASIBILITY

It is concerned with the returns or benefits of the organization are likely to derive from investment in the system. Estimated costs of new system development and operation must be balanced against projected tangible as well as intangible benefits.

3.2 OPERATIONAL FEASIBILITY

It is an evaluation to determine whether a system is operationally acceptable. Two important dimensions to be accessed are ability and motivation to use the system.

3.3 MANAGERIAL FEASIBILITY

It determines whether a proposed system will be acceptable to the people or not. It also determines from the management.

CHAPTER 4

SYSTEM ANALYSIS

- The system runs on apache server. So, it is needed that the server must have an apache server version 2.0 available. XAMPP is used for running apache server and MYSQL server
- MySQL database has been used for storing the data of the website.
- HTML and PHP has been used for creating the layout of the web application.
- CSS has been used for creating and designing of the webpage.

4.1 HARDWARE INTERFACE

- System : Pentium IV 2.4 GHz, Intel Core i3, Core i7
- Hard Disk : 40 GB and higher.
- RAM : 256 Mb and higher.
- SPEED : 1.8Ghz
- Keyboard and Monitor

4.2 SOFTWARE INTERFACE

- Operating System : Windows 7
- Database used : MySQL
- Language : HTML/Java Script

CHAPTER 5

DATABASE DESIGN

5.1 RELATIONAL MODEL

Relational data model is the primary data model, which is used widely around the world for data storage and processing. This model is simple and it has all the properties and capabilities required to process data with storage efficiency.

In the relational model, all data must be stored in relations (tables), and each relation consists of rows and columns. Each relation must have a header and body. The header is simply the list of columns in the relation. The body is the set of data that actually populates the relation, organized into rows. You can extrapolate that the junction of one column and one row will result in a unique value - this value is called a tuple.

The second major characteristic of the relational model is the usage of keys. These are specially designated columns within a relation, used to order data or relate data to other relations. One of the most important keys is the primary key, which is used to uniquely to identify each row of data. To make querying for data easier, most relational databases go further and physically order the data by the primary key. Foreign keys relate data in one relation to the primary key of another relation. Besides defining how the data are to be structured as discussed above, the relational model also lays down a set of rules to enforce data integrity, known as integrity constraints. It also defines how the data are to be manipulated (relational calculus). In addition, the model defines a special feature termed normalization to ensure efficient data storage.

1) **ENTITY**

   An entity is an object that has its existence in the real world. It includes all those "things" about which data is collected. An entity may be a tangible object such as a student, a place or a part. It may also be non-tangible such as an event, a job title or a customer account. For example, if we say that a customer book room then it means

customer and room both are entities. Diagrammatically, entities are represented in rectangles.

2) AN ENTITY SET

It is a set of entities of the same type that share the same properties, or attributes. The set of all persons who are patients at some hospital is example can be defined as the entity set customer.

3) ATTRIBUTES

Attributes are units that describe the characteristics or properties of entities. In a database, entities are represented by tables and attributes by columns. For example, a customer entity might have numerous attributes such as id, first name, last name address, and phone no, the room entity may have attributes like room no, room type and etc., They are drawn in elliptical shapes along with the entity rectangles.

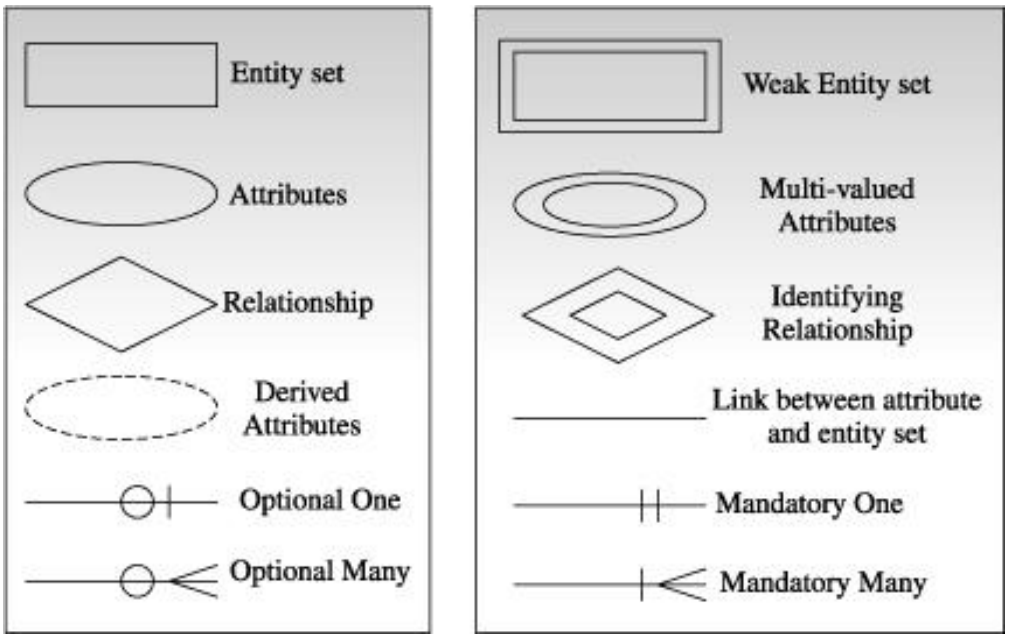

Figure 5.1.1 Database Design

5.2 ENTITY RELATIONSHIP DIAGRAM

Entity Relationship Diagrams (ERDs) illustrate the logical structure of databases. It is a piece of data-an object or concept about which data is stored. The below ERDs Represents the Hospital Database.

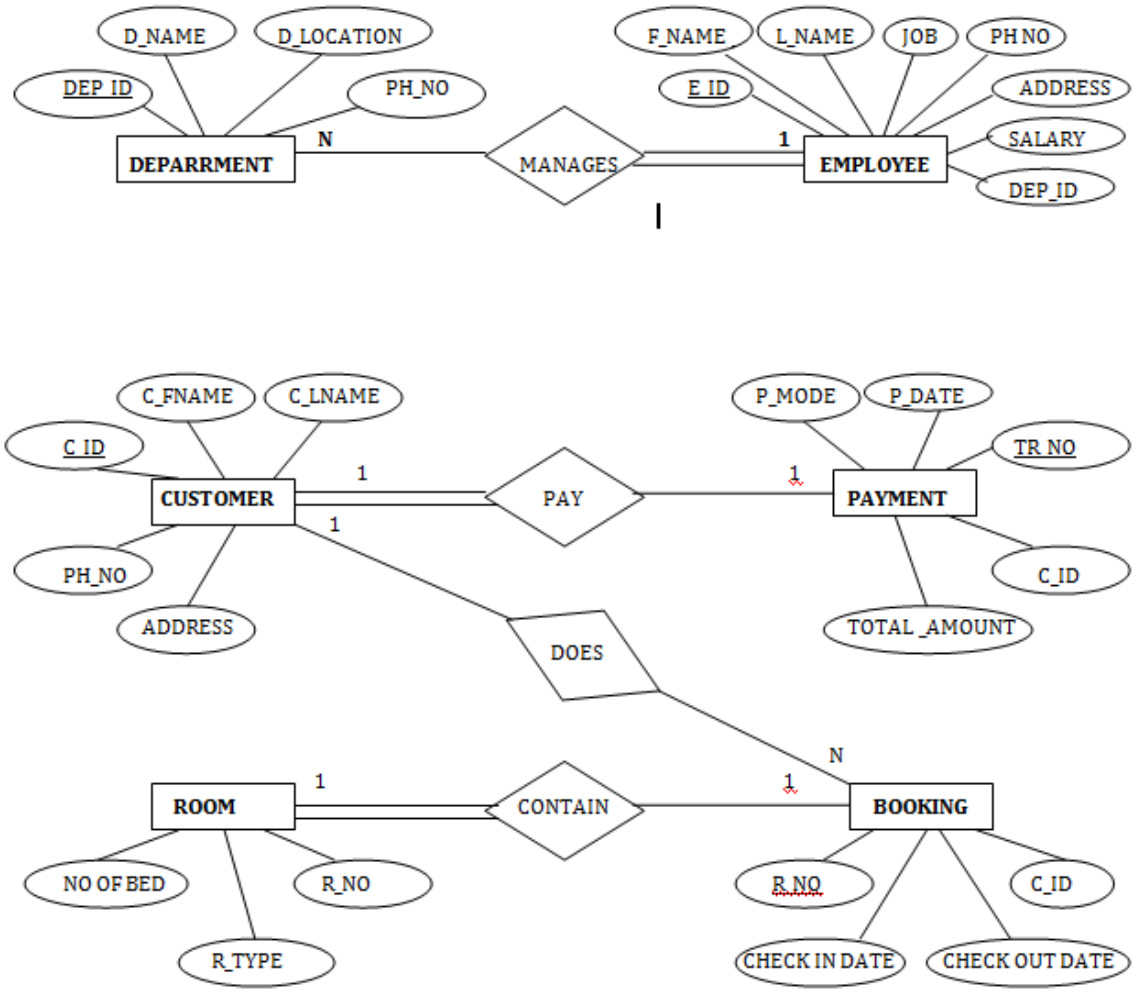

Figure 5.2.1 Entity Relationship Diagram

5.3 SCHEMA

A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams.

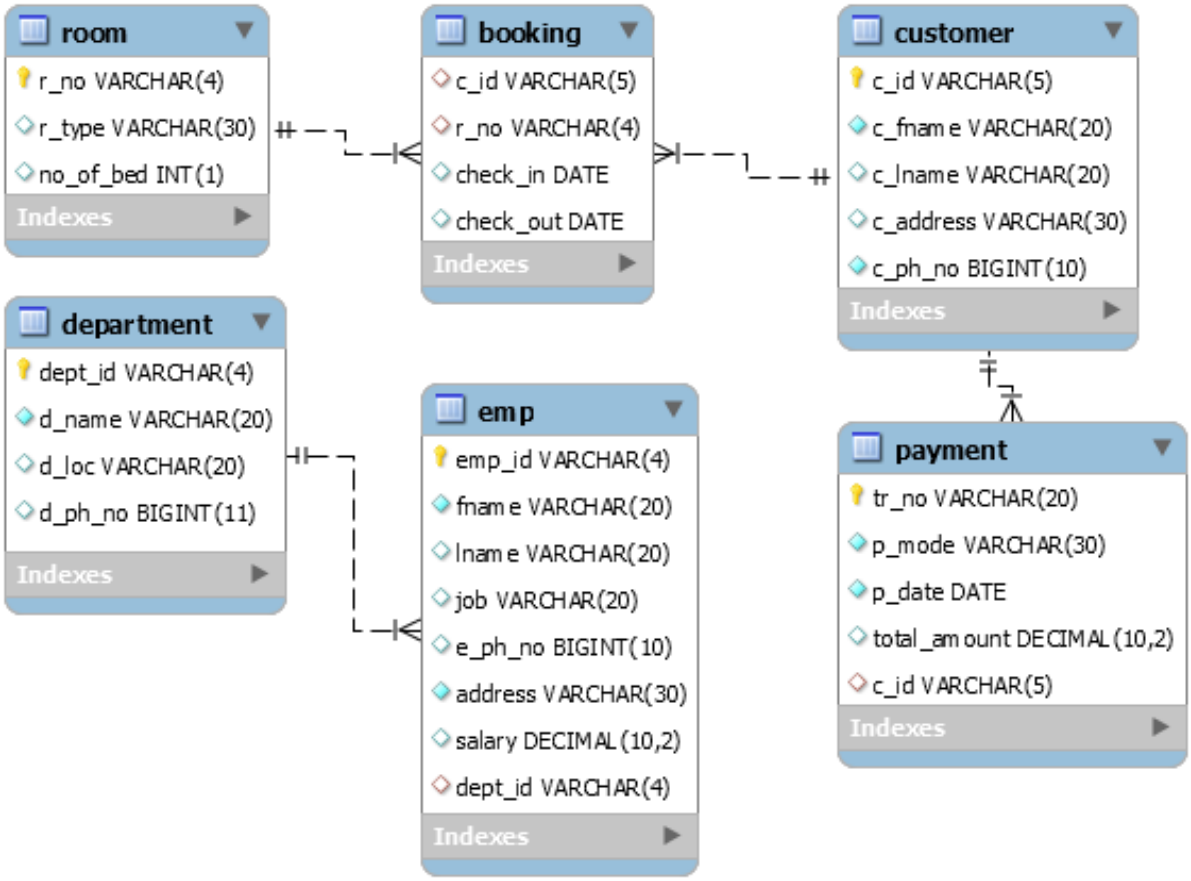

Figure 5.3.1 Schema

CHAPTER 6

IMPLEMENTATION

The modules of this application include:

- CUSTOMERS - Used for storing the customer information such as entry, check in , check out etc.,
- ROOM - Stores the rooms details such as no of beds of rooms and type of room etc.,
- BOOKING – Stores the reservation details along with check in date check out date.
- PAYMNET – Stores payment id, total amount, date of payment, and mode of payment.
- DEPARTMNET – Used to store how many employees ha sassing for each department designation and description of each department
- EMPLOYEES– Stores the Details of employees, salary of employee along with department id so that easily employee could to tacked ,which employee is under which department.

6.1 TABLE CREATION

1) DEPARTMENT TABLE

create table department(dept_id varxchar(4) primary key,d_name varchar(20) not null,d_loc varchar(20) ,d_ph_no bigint(11) unsigned zerofill);

desc department;

;

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	dept_id	varchar(4)			No	None	
2	d_name	varchar(20)			No	None	
3	d_loc	varchar(20)			Yes	NULL	
4	d_ph_no	bigint(11)		UNSIGNED ZEROFILL	Yes	08022146133	

Figure 6.1.1 department table structure

2) EMPLOYEE TABLE

create table emp(emp_id varchar(4) primary key,fname varchar(20) not nul l,lname varchar(20),designation varchar(30),e_ph_no bigint(10),address varchar(30) not null, salary decimal(10,2),dept_id varchar(4),foreign key(dept_id) references dep artment(dept_id) on delete cascade on update cascade);

desc emp;

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	emp_id	varchar(4)			No	None	
2	fname	varchar(20)			No	None	
3	lname	varchar(20)			Yes	NULL	
4	job	varchar(20)			Yes	NULL	
5	e_ph_no	bigint(10)			Yes	NULL	
6	address	varchar(30)			No	None	
7	salary	decimal(10,2)			Yes	NULL	
8	dept_id	varchar(4)			Yes	NULL	

Figure 6.1.2 employee table structure

3) CUSTOMER TABLE

create table customer(c_id varchar(5) primary key,c_fname varchar(20) not null,c_lname varchar(20),c_address varchar(30),c_ph_no bigint(10) not null);

desc customer;

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	**c_id**	varchar(5)			No	*None*	
2	**c_fname**	varchar(20)			No	*None*	
3	**c_lname**	varchar(20)			Yes	*NULL*	
4	**c_address**	varchar(30)			Yes	*NULL*	
5	**c_ph_no**	bigint(10)			No	*None*	

Figure 6.1.3 customer table structure

4) ROOM TABLE

create table room(r_no varchar(4) primary key,r_type varchar(30) default 'general',no_of_bed int(1) default 1);

desc room;

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	**r_no**	varchar(4)			No	*None*		Change
2	**r_type**	varchar(30)			Yes	general		Change
3	**no_of_bed**	int(1)			Yes	1		Change

Figure 6.1.4 room table structure

5) PAYMENT TABLE

create table payment(tr_no varchar(20) primary key,p_mode varchar(30) not null,p_date date not null,total_amount decimal(10,2),c_id varchar(5),foreign key(c_id) references customer(c_id)on delete cascade on update cascade);

desc payment;

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	tr_no	varchar(20)			No	None		Change
2	p_mode	varchar(30)			No	None		Change
3	p_date	date			No	None		Change
4	total_amount	decimal(10,2)			Yes	NULL		Change
5	c_id	varchar(5)			Yes	NULL		Change

Figure 6.1.5 payment table structure

6) BOOKING TABLE

create table booking(c_id varchar(5),r_no varchar(4),check_in_date date not null,check_out_date date not null,foreign key(c_id) references custom
er(c_id) on delete cascade on update cascade,foreign key(r_no) references room(r_no) on delete cascade on update cascade);

desc booking;

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	c_id	varchar(5)			Yes	*NULL*		Change
2	r_no	varchar(4)			Yes	*NULL*		Change
3	check_in	date			Yes	*NULL*		Change
4	check_out	date			Yes	*NULL*		Change

Figure 6.1.6 booking table structure

6.2 DDL COMMANDS

1) Add a primary key

Alter table customer add primary key(c_id) not null;

2) Delete a row from patient table

Delete from customer where c_id=’C0001’;

3) Insert a row into patient table

insert into customer values(‘C0001’,'RAKESH','SHARMA',987654321,
'BANGALORE');

4) Change the name of employee table to emp table

rename table project.employee to project.emp;

5) Drop the emp table

Drop table emp;

6) Add unique constraint to the department name

Alter table department add constraint uq unique(d_name);

7) set not null constrain to d_name of department table

alter table department modify d_name varchar(20) not null;

8) Change the column name designation to job from employee table

alter table emp change designation job varchar(20);

6.3 AGGREGATION

1) Display the minimum, maximum, average salaries of doctors

Select min(dsalary) as minimum_salary,max(dsalary) as maximum_salary,avg (dsalary) as average_salary from employee;

2) Display how many departments are there in Hotel Database

select count(d_id) from department;

3) Display total salary, no of employees, average of salary, highest salary, and lowest salary using aggregate functions

select sum(salary) as 'total Salary',count(*) as 'no_of_emplyees' ,max(salary) ,min(salary) as 'lowest salary' ,avg(salary) as 'average of salary' from employee;

4) Display the minimum, maximum and average payments of customers

select min(total_amount), max(total_amount), avg(total_amount) from payment;

5) Display the no of customers

select count(c_id) from customer;

6) **Display the no customers according to their address**

select age,address,count(c_id) from customer group by address order by address;

7) **Display the no of employees working for each department**

select d_id,count(e_id) from employee group by d_id;

6.4 JOINS

1) **Display the employee name and departments name of employee and department table**

select d.e_name,concat(‘fname,' ',lname) as 'employee name' from department d,employee do where d.deptid=do.deptid;

2) **Display the details of employee and department**

select d.fname,d.designation,dep.deptname,dep.deploc from employee d,department dep where d.deptid=dep.deptid;

3) **display the customer name from customer table and check in date from booking table who have checked in, in July month using joins**

select concat(c_fname,c_lname) as 'customer name',check_in from customer c,booking b where c.c_id=b.c_id and check_in like '___-07-__';

4) **Display the customer full name as customer name and room full name as room type**

```
select concat(c_fname,c_lname) as 'customer name',check_in from
customer c,booking b where c.c_id=b.c_id;
```

6.5NESTED AND CORRELATED QUERIES

1) **display the details of available rooms in hotel using nested queries**

```
select *from room where r_no not in(select r_no from booking);
```

2) **List out the department details which does not contain any employee**

```
select *from department where d_id not in(select d_id from employee);
```

3) **Display the department details who are not treating any employee**

```
select *from department where d_id not in(select d_id from employee);
```

4) **Display the details of departments who are working and working in any department using exists and not exists**

```
select *from department dep where exists(select *from employee e where
e.d_id=dep.d_id);
```

```
select *from department dep where not exists(select *from employee e where
e.d_id=dep.d_id);
```

5) **display the details of available rooms in hotel using nested queries**

```
select * from room where r_no not in(select r_no from booking);
```

6) **display the customer name and paid amount which is greater then average of paid amount using correlated queries**

```
select concat(c_fname,' ',c_lname) as 'customer name',total_amount from
customer c, payment p where c.c_id=p.c_id and p.total_amount>(select
avg(total_amount) from payment );
```

6.6 VIEWS

1) **Create view to list employees how are from bangalore**

```
create view v1 as select * from employee where address =”Bangalore”;

select * from v1;
```

2) **Create view to list minimum, maximum, average, total salary of employee**

```
create view v12 as select min(salary) as minimum_cost, max(salary) as
maximum_cost, avg(salary) as average_cost , sum(salary) as total_cost from
employee;

select *from v2;
```

3) **Create view to display name of customer and type of room reserved**

```
create view v3 as select concat(c_fname,' ',c_lname) as “CustomerName”
,r_type from customer c, room r, booking  b, where c.c_id=b.c_id and
r.r_no=b.r_no;
select *from v3;
```

6.7 TRIGGERS

1) **Create trigger to throw an error message if salary of employee is greater then 30000**

   Delimiter $

   Create trigger t1 before insert on employee begin if new.salary>30000 then signal sqlstate ‘45000’ set message_text=”enter correct salary of employee”; end if; end ; $

   Insert employee details salary greater then 30000

2) **Create trigger to keep track of, no of booking done in booking table.**

   Delimiter $

   create trigger t2 after insert on booking beging delete from track: insert into track select count(c_id) from booking; end ;$

   insert values in booking table and checgk in track table u will be able to see the count of booking done

3) **Create trigger to set ph no to 99999999999 if new ph no of department is inserted or updated as 0.**

   Delimiter $

   create trigger t3 before insert on department for each row begin if new.ph_no=0 then set new.ph_no=9999999999; end if; end; $

   Insert ph no as 0 in department table to get the result

Trigger	Event	Table	Statement	Timing	Created	sql_mode	Definer	character_set_client	collation_connection	Database Collation
t1	INSERT	booking	begin delete from track; insert into track select count(c_id) from booking; end	AFTER		NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION	root@localhost	latin1	latin1_swedish_ci	latin1_swedish_ci

Figure 6.7.1 Triggers

CHAPTER 7

SNAPSHOTS

1) Front End of Hospital Database

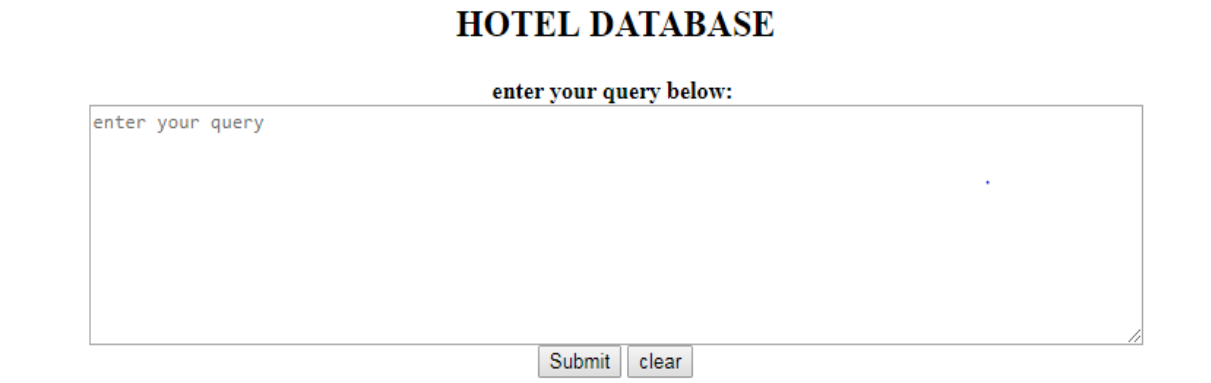

Figure 7.1 Front end of hotel database

2) Employee Details

emp_id	fname	lname	job	e_ph_no	address	salary	dept_id
E001	salman	khan	manager	9876543210	bangalore	45000.00	D006
E002	shahid	kapur	electrician	9876543211	mumbai	15000.00	D004
E003	karina	kaif	receptionist	9876543212	mumbai	20000.00	D008
E004	suhail	khan	technician	9876543213	chennai	25000.00	D004
E005	varun	k	help_line	9876543214	bangalore	15000.00	D005
E006	varun	dhawan	cook	9876543215	bangalore	25000.00	D003
E007	vijay	kumar	billing	9876543216	mumbai	15000.00	D001
E008	raj	kumar	guard	9876543217	mumbai	20000.00	D002
E009	akshay	kumar	waiter	9876543218	chennai	15000.00	D007
E010	ajay	raj	floor cleaner	9876543219	bangalore	15000.00	D009
E011	ram	raj	waiter	9876543218	chennai	15000.00	D007
E013	ram	ling	waiter	9876543210	bangalore	16000.00	D007
E014	ram	ling	waiter	9876543210	bangalore	16000.00	D007

Figure 7.2 employee details

3) customer Details

c_id	c_fname	c_lname	c_address	c_ph_no
C0001	raghu	ram	bangalore	9123456780
C0002	raja	ram	bangalore	9123456781
C0003	mohan	roy	bangalore	9123456782
C0004	bhagat	singh	bangalore	9123456783
C0005	kapil	singh	mumbai	9123456784
C0006	kapil	sharma	mumbai	9123456785
C0007	ajay	sharma	mumbai	9123456786
C0008	vijay	k	chennai	9123456787
C0009	anna	singh	chennai	9123456788
C0010	ajay	sharma	chennai	9123456789
C0011	vijai	sharma	chennai	9123456790
C0012	rahul	sharma	chennai	9123456791
C0013	virat	sharma	delhi	9123456792
C0014	varun	sharma	delhi	9123456793

Figure 7.3 customer details

4) Department Details

dept_id	d_name	d_loc	d_ph_no
D001	accounts	A_block	08022146131
D002	security	A_block	08022146132
D003	food production	B_block	08022146134
D004	maintenance	c_block	08022146135
D005	general_help	D_block	08022146133
D006	managing	A_block	08022146138
D007	food supplying	A_block	08022146136
D008	reception	A_block	08022146137
D009	house keepers	c_block	08022146130

Figure 7.4 Department Details

5) Booking Details

c_id	r_no	check_in	check_out
C0002	R002	2018-01-01	2018-01-05
C0004	R013	2018-02-20	2018-03-05
C0001	R009	2018-01-01	2018-01-28
C0003	R010	2018-02-10	2018-02-25
C0011	R011	2018-07-25	2018-08-10
C0012	R014	2018-07-28	2018-08-08
C0005	R005	2018-03-15	2018-04-10
C0006	R003	2018-04-15	2018-05-01
C0007	R006	2018-04-20	2018-05-01
C0008	R008	2018-05-20	2018-06-01
C0009	R015	2018-06-25	2018-07-20
C0010	R004	2018-07-20	2018-08-01
C0013	R001	2018-08-18	2018-08-27
C0014	R007	2018-09-10	2018-09-20
C0014	R017	2018-08-03	2018-09-02
C0014	R016	2018-08-01	2018-08-02

Figure 7.5 Booking details

6) Rooms Details

RNO	RNAME	RDESCRIPTION	PID
A111	CASUALTY	Normal consulation	101
A112	MATERNITY	Giving Birth	102
A113	DISPENSARY	Medicines	103
A114	EMERGENCY	Accidents	104
A115	HOUSEKEEPING	Cleaning	105
A116	ICU	Intensive care	106
A117	SURGERY	Doing Surgeries	107
A118	NORMAL WARD	Patients Stay	108
A119	NURSERY	Babies Stay	109
A120	PADDED CELL	Mental Patients	110

Figure 7.6 Rooms details

CHAPTER 8

CONCLUSION

Finally, I Conclude by saying that using this Hotel Database system we can easily reduce the manual work and can generate more securable and more efficient database which can record various data related to hotel infrastructure and also by using this system man work is also reduced by that was previously done using paper and forms. This project has more scope in the future and can be integrated further. This project is successfully implemented with all the features mentioned earlier. Deployment of this application will help the hospital management to reduce the unnecessary wastage of time in doing manually. Therefore, we are successfully able to reach the goals and target of the project. Thus, keeping the working of the manual system as the basis of our system. We have developed an automated version of the manual system, named as Hotel Database. This project Hotel Database aimed to develop and maintain the day-to-day state of bookings, payments paid of customerss, list of employees ,reports generation, and do many more works automatically.

In a nutshell it can be summarized that the future scope of the project circles around maintaining information regarding:

- We can add a printer.
- We can give for advance front end facilities.
- We can host the platform on online servers to make it accessible worldwide.
- Create a master and a slave database structure to reduce the overload of the database queries.
- Implement the backup mechanism for taking backup of codebase and database on regular basis on different servers

CHAPTER 9

BIBLIOGRAPHY

- https://en.wikipedia.org/wiki/hotel_database
- https://www.youtube.com/hotel_database
- https://www.slideshare.com/hotel_management
- https://erdplus.com
- https://www.geeksforgeeks.org/relational-model/
- https://www.tutorialspoint.com/dbms/dbms_data_schemas.html