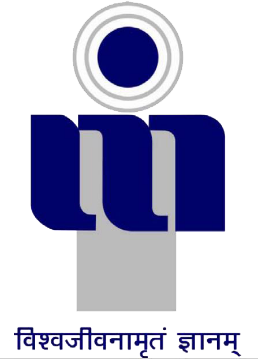
**ABV-INDIAN INSTITUTE OF INFORMATION TECHNOLOGY &MANAGEMENT, GWALIOR**

****

**HOTEL RESERVATION SYSTEM**

**PROJECT WORK**

*Submitted by:*

Evi Agarwal (2014-IPG-108)

Kriti Singhal (2014-IPG-050)

Lekhni Agarwal (2014-IPG-051)

Shivani Mandeliya (2014-IPG-083)

Under the guidance of

Dr. Vinay Kumar Singh

**Contents:**

* Introduction
* Objective
* Hotel Schema
* Entities and Attributes
* Entities and their Description
* Relationships and their Description
* Entities’ Description
* E-R Diagram
* E-R Model to Relational Model
* Normalization
* 1st Normal Form
* 2nd Normal Form
* 3rd Normal Form
* Boyce Codd Normal Form

**INTRODUCTION**

This project introduces the hotel reservation system.The term hotel reservation system automates the task of room reservation, hall booking, bill generation, checks room availability and maintains the record of various other facilities provided by the hotel.

It explains how room booking is done in a hotel. Customers should be able to know the availability of the rooms on a particular date. They should be able to reserve the available rooms according to their need in advance. It helps in maintaining the record of the guests and reserve rooms beforehand.

The main aim of this project is to implement an efficient database system to manage all the boarding and lodging activities of a hotel.

**OBJECTIVE**

The goal of this project is to introduce the efficient, consistent and computerized hotel management system to the society of the hotels and let them get the opportunity to use the most happening technology to prosper their business. Apart from this, it aims to provide the advanced security, thus protecting the confidentiality of all information stored in the system and limiting the access control.

**Hotel Schema**

**ENTITIES AND ATTRIBUTES:**

* **Guest: <**reservation\_folio\_id, contact\_no, email\_id, arr\_frm, prcdng\_to, prps\_of\_visit, no\_of\_members>.
* Guest can be specialized into two types: family or corporate.

**Family:**<head\_id, head\_name, address, no\_of\_children, profession, no\_of\_adults, age, gender>.

**F\_members: <**head\_id**,** name, age, gender>.

**Corporate:**<company\_id, company\_name, address>.

**Corp\_members:** <company\_id, contact\_no company\_name, age, gender, designation>.

* **Room: <**room\_no, room\_type, rate, status>
* **Food:** <pdct\_name, pdct\_price, food\_type>
* **Offers:**<package\_type, description>
* **Facilities:**<facility\_type, facility\_rate>
* **Employee:**<emp\_id, emp\_name, post, emp\_mb\_no, facility\_type>
* **Bill:**<bill\_no, bill\_date, room\_charges, food\_charges,ser\_tax, vat, lux\_tax, grand\_total>

Entities and their Description:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity** | **Type of Entity** | **Primary Key** | **Foreign Key** | **Discriminator** |
| Guest | Strong | Reservation\_folio\_id | --- | --- |
| Family | Strong | Head\_id | --- | --- |
| F\_members | Weak | Head\_id, name | Head\_id | name |
| Corporate | Strong | Company\_id | --- | --- |
| Corp\_members | Weak | Company\_id, contact\_no | Company\_id | Contact\_no |
| Room | Strong | Room\_no | --- | --- |
| Food | Strong | Pdct\_name | --- | --- |
| Offers | Strong | Package\_type | --- | --- |
| Facilities | Strong | Facility\_type | --- | --- |
| Employee | Strong | Emp\_id | Facility\_type | --- |
| Bill | Strong | Bill\_no | --- | --- |

**Relationships and their description**

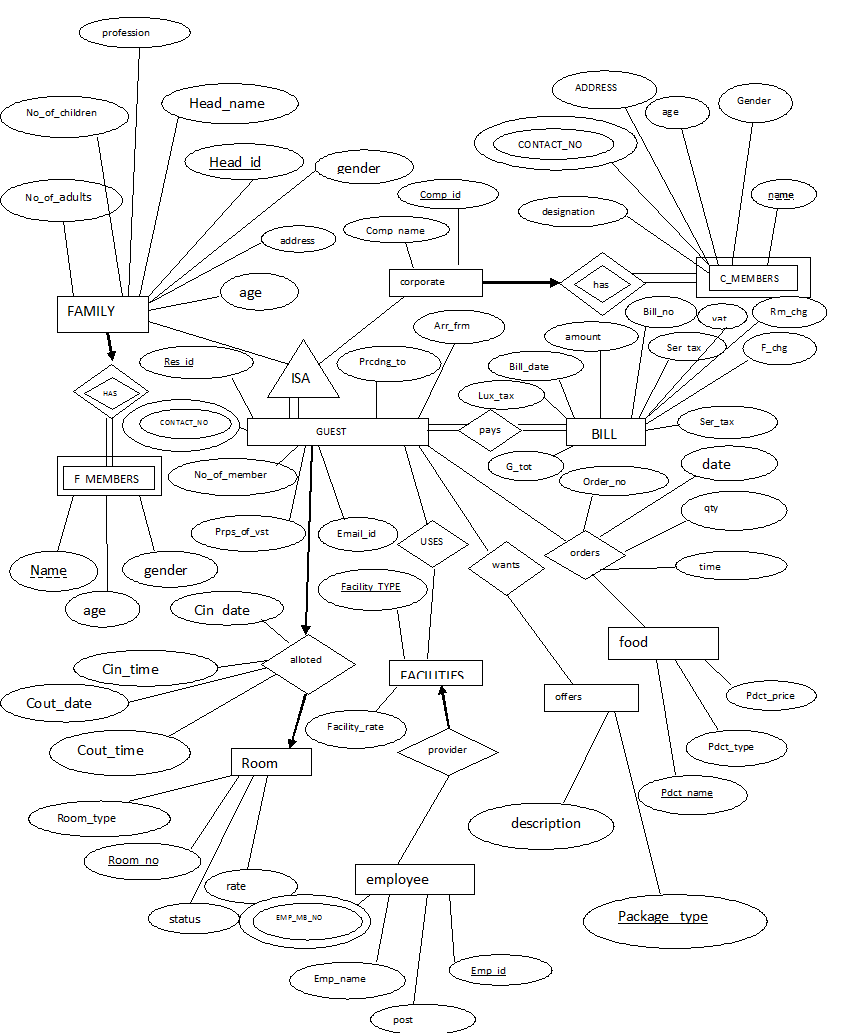
*We have assumed all the relationships to be binary.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entity\_1** | **Entity\_2** | **Relationship** | **Descriptive**  **Attributes** | **Mapping**  **Cardinality** | **Participation** |
| Guest | Room | Alloted | cin\_date, cout\_date, cin\_time, cout\_time | One to one | Partial |
| Guest | Food | Orders | date, time, qty, order\_no | Many to many | Partial |
| Guest | Bill | Pays | payment\_method | One to one | Total |
| Guest | Facilities | Uses | - | Many to many | Partial |

**Entities description**

* **Food-type:** Indian, Italian, Chinese, South-Indian, Thai, Continental, Dessert, Cocktail, Sea-food, Non-veg and beverages.
* **Facilities:** Restaurant, Baby-sitting, laundary, travel assistance, currency exchange, cab hire service, gym, spa, doctor on call, pool, buffet, bar, business center and Wi-Fi.
* **Room type:** Single AC Rooms, Single non AC rooms, Double AC rooms, Double non AC rooms, Junior suite, Superior Suite, Executive Suite, Grande suite and Family Suite.
* **Offers:**

1. Discount Offer: 10% discount on suite on 5 day and 4 night stay.
2. Extended stay offer: Stay for minimum 7 nights, 6 days and avail breakfast, airport transfers by cabs, laundary, internet and a lot more.
3. Advance purchase offer: Book 14 days, 13 nights ahead and avail our special advance purchase rate with full prepayment.
4. Business package: Stay for minimum 3 nights and avail breakfast, one way airport transfer, 2 pieces of laundary each day and internet.
5. Family package: Stay with comfort by availing a complementary room for 2 kids upto 12 years of age.

E-R DIAGRAM

**E-R Model to Relational Model**

**Entities:**

Entity 1: **Guest**[tables\GUEST.xlsx](tables/GUEST.xlsx)

Entity 2: **Family**[tables\family.xlsx](tables/family.xlsx)

Entity 3: **F\_members**[tables\F\_members.xlsx](tables/F_members.xlsx)

Entity 4: **Corporate**[tables\Corporate.xlsx](tables/Corporate.xlsx)

Entity 5: **Corp\_members**[tables\corp\_members.xlsx](tables/corp_members.xlsx)

Entity 6: **Food**[tables\Food Relation.xlsx](tables/Food%20Relation.xlsx)

Entity 7: **Bill**[tables\BILL.xlsx](tables/BILL.xlsx)

Entity 8: **Offers**[tables\Offers Relation.xlsx](tables/Offers%20Relation.xlsx)

Entity 9: **Room**[tables\Room Relation.xlsx](tables/Room%20Relation.xlsx)

Entity 10: **Employee**[tables\employee.xlsx](tables/employee.xlsx)

Entity 11: **Facility**[tables\facility.xlsx](tables/facility.xlsx)

**Relationships:**

Relationship 1: **Alloted**<Guest is alloted room.>

[tables\guest\_alloted\_room.xlsx](file:///C:\Users\Kriti%20Singhal\Desktop\dbms%20proj\tables\guest_alloted_room.xlsx)

Relationship 2: **Order <**Guest orders food.>

[tables\guest\_orders\_food.xlsx](tables/guest_orders_food.xlsx)

Relationship 3: **Pays**<Guest pays bill.>

[tables\guest\_pays\_bill.xlsx](tables/guest_pays_bill.xlsx)

Relationship 4: **Uses**<Guest uses Facilities.>

[tables\guest\_uses\_facilities.xlsx](tables/guest_uses_facilities.xlsx)

**NORMALIZATION**

**1ST Normal Form**

First normal form is defined in the definition of relations (tables) itself. This rule defines that all the attributes in a relation must have atomic domains. The values in an atomic domain are indivisible units.

**Entities:**

Entity 1: **Guest** [**INF\GUEST.xlsx**](INF/GUEST.xlsx)

Entity 2: **Family** [**INF\family.xlsx**](file:///C:\Users\Kriti%20Singhal\Desktop\dbms%20proj\INF\family.xlsx)

Name :- <first\_name, middle\_name, last\_name>

Address :- <house\_no\_or\_name, area, city, state>

Entity 3: **F\_members**[**INF\F\_members.xlsx**](file:///C:\Users\Kriti%20Singhal\Desktop\dbms%20proj\INF\F_members.xlsx)

Name :- <first\_name, last\_name >

Entity 4: **Corporate**[**INF\Corporate.xlsx**](file:///C:\Users\Kriti%20Singhal\Desktop\dbms%20proj\INF\Corporate.xlsx)

Address :- <factory\_name, area, city, state>

Entity 5: **Corp\_members**[**INF\corp\_members.xlsx**](file:///C:\Users\Kriti%20Singhal\Desktop\dbms%20proj\INF\corp_members.xlsx)

Name: -<first\_name, middle\_name, last\_name>

Entity 6: **Food** [**INF\Food Relation.xlsx**](INF/Food%20Relation.xlsx)

Entity 7: **Bill** [**INF\BILL.xlsx**](INF/BILL.xlsx)

Entity 8: **Offers** [**INF\Offers Relation.xlsx**](INF/Offers%20Relation.xlsx)

Entity 9: **Room** [**INF\Room Relation.xlsx**](INF/Room%20Relation.xlsx)

Entity 10: **Employee**[**INF\employee.xlsx**](file:///C:\Users\Kriti%20Singhal\Desktop\dbms%20proj\INF\employee.xlsx)

Name: -<first\_name, middle\_name, last\_name>

Entity 11: **Facility**[**INF\facility.xlsx**](INF/facility.xlsx)

**RELATIONSHIPS:**

Relationship 1: **Alloted**<Guest is alloted room.>

[INF\guest\_alloted\_room.xlsx](file:///C:\Users\Kriti%20Singhal\Desktop\dbms%20proj\INF\guest_alloted_room.xlsx)

Relationship 2: **Orders**<Guest orders food.>

[INF\guest\_orders\_food.xlsx](INF/guest_orders_food.xlsx)

Relationship 3: **Pays**<Guest pays bill.>

[INF\guest\_pays\_bill.xlsx](INF/guest_pays_bill.xlsx)

Relationship 4: **Uses**<Guest uses facilities.>

[INF\guest\_uses\_facilities.xlsx](INF/guest_uses_facilities.xlsx)

**2nd NORMAL FORM**

In second normal form, every non-prime attribute should be fully functionally dependent on prime key attribute. That is, if X → A holds, then there should not be any proper subset Y of X, for which Y → A also holds true.

**Entities:**

Entity 1: **Guest**[2 NF\GUEST.xlsx](file:///C:\Users\Kriti%20Singhal\Desktop\dbms%20proj\2%20NF\GUEST.xlsx)

The FD is as follows:

*Reservation\_folio\_id 🡪 contact\_no, email\_id, no\_of\_members, arr\_frm, prcdng\_frm, prps\_of\_visit, package\_type*

Since, no partial functional dependency exist in this relation, we do not decompose it further.

Entity 2: **Family**[2 NF\family.xlsx](file:///C:\Users\Kriti%20Singhal\Desktop\dbms%20proj\2%20NF\family.xlsx)

The FD is as follows:

*Head\_id 🡪 first\_name, middle\_name, last\_name, no\_of\_children, no\_of\_adults, gender, age, profession, house\_no\_or\_name, area, city, state.*

Since, no partial functional dependency exist in this relation, we do not decompose it further.

Entity 3: **F\_members**[2 NF\F\_members.xlsx](file:///C:\Users\Kriti%20Singhal\Desktop\dbms%20proj\2%20NF\F_members.xlsx)

The FD is as follows:

*Head\_id, first\_name, middle\_name, last\_name 🡪 age, gender.*

Since, no partial functional dependency exist in this relation, we do not decompose it further.

Entity 4: **Corporate**

The FD is as follows:

*Company\_id 🡪 company\_name, factory\_name, area, city, state*

Since, no partial functional dependency exist in this relation, we do not decompose it further.

Entity 5: **Corp\_members**

In this relation, partial functional dependency exists between contact no and name i.e.

*Contact\_no 🡪First\_name, middle\_name, last\_name*

Hence, we need to decompose the relation corporate into following 2 relations:

**Corp\_members 1:**contact\_no, first\_name, middle\_name, last\_name

[2 NF\corp\_members1.xlsx](file:///C:\Users\Kriti%20Singhal\Desktop\dbms%20proj\2%20NF\corp_members%201.xlsx)

**Corp\_members 2:**company\_id, contact\_no, age, gender, designation

[2 NF\corp\_members2.xlsx](2%20NF/corp_members%202.xlsx)

Entity 6: **Food**[2 NF\FoodRelation.xlsx](file:///C:\Users\Kriti%20Singhal\Desktop\dbms%20proj\2%20NF\Food%20Relation.xlsx)

The FD is as follows:

*Pdct\_name 🡪 pdct\_price, food\_type.*

Since, no partial functional dependency exist in this relation, we do not decompose it further.

Entity 7: **Bill**[2 NF\BILL.xlsx](file:///C:\Users\Kriti%20Singhal\Desktop\dbms%20proj\2%20NF\BILL.xlsx)

The FD is as follows:

*Bill\_no 🡪 bill\_date, room\_charges, ser\_tax, food\_charges, vat, lux\_tax, grand\_total.*

Since, no partial functional dependency exist in this relation, we do not decompose it further.

Entity 8: **Offers** [2 NF\Offers Relation.xlsx](file:///C:\Users\Kriti%20Singhal\Desktop\dbms%20proj\2%20NF\Offers%20Relation.xlsx)

The FD is as follows:

*Package\_type 🡪 description*

Since, no partial functional dependency exist in this relation, we do not decompose it further.

Entity 9: **Room**[2 NF\Room Relation.xlsx](file:///C:\Users\Kriti%20Singhal\Desktop\dbms%20proj\2%20NF\Room%20Relation.xlsx)

The FD is as follows:

*Room\_no 🡪 room\_type, rate, status.*

Since, no partial functional dependency exist in this relation, we do not decompose it further.

Entity 10: **Employee**[**2 NF\employee.xlsx**](2%20NF/employee.xlsx)

The FD is as follows:

*Emp\_id 🡪 emp\_name, emp\_mb\_no, post, facility\_type*

Since, no partial functional dependency exist in this relation, we do not decompose it further.

Entity 11: **Facility**[2 NF\facility.xlsx](file:///C:\Users\Kriti%20Singhal\Desktop\dbms%20proj\2%20NF\facility.xlsx)

The FD is as follows:

*Facility\_type🡪 facility\_rate*

Since, no partial functional dependency exist in this relation, we do not decompose it further.

**RELATIONSHIPS:**

Relationship 1: **Alloted**[2 NF\guest\_alloted\_room.xlsx](file:///C:\Users\Kriti%20Singhal\Desktop\dbms%20proj\2%20NF\guest_alloted_room.xlsx)

The FD is as follows:

*Reservation\_folio\_id, Room\_no 🡪 cin\_date, cin\_time, cout\_date, cout\_time*

Since, no partial functional dependency exist in this relation, we do not decompose it further.

Relationship 2: **Orders**[**2 NF\guest\_orders\_food.xlsx**](file:///C:\Users\Kriti%20Singhal\Desktop\dbms%20proj\2%20NF\guest_orders_food.xlsx)

The FD is as follows:

*Reservation\_folio\_id, pdct\_name 🡪 order\_no, date, time, qty.*

Since, no partial functional dependency exist in this relation, we do not decompose it further.

Relationship 3: **Pays**[2 NF\guest\_pays\_bill.xlsx](file:///C:\Users\Kriti%20Singhal\Desktop\dbms%20proj\2%20NF\guest_pays_bill.xlsx)

The FD is as follows:

*Reservation\_folio\_id, bill\_no 🡪 payment\_method*

Since, no partial functional dependency exist in this relation, we do not decompose it further.

Relationship 4: **Uses**[2 NF\guest\_uses\_facilities.xlsx](file:///C:\Users\Kriti%20Singhal\Desktop\dbms%20proj\2%20NF\guest_uses_facilities.xlsx)

Since, no partial functional dependency exist in this relation, we do not decompose it further.

**3rd NORMAL FORM**

For a relation to be in Third Normal Form, it must be in Second Normal form and no non-prime attribute is transitively dependent on prime key attribute.

**Entities:**

Entity 1: **Guest**[3 NF\GUEST.xlsx](file:///C:\Users\Kriti%20Singhal\Desktop\dbms%20proj\3%20NF\GUEST.xlsx)

Since there is no transitive dependency among any of the attributes, thus it is already in 3rd NF. Hence no further decomposition is required.

Entity 2: **Family**

In this relation, following transitive functional dependency exists-:

1. *Head\_id 🡪First\_name, middle\_name, last\_name 🡪 house\_no\_or\_name, area, city, state.*
2. *Head\_id 🡪 first\_name, middle\_name, last\_name 🡪 profession*

Hence, we need to decompose the relation employee into following 3 relations:

**Family 1:** Head\_id, First\_name, middle\_name, last\_name, no\_of\_children, no\_of\_adults, age, gender.

[3 NF\family\_1.xlsx](3%20NF/family_1.xlsx)

**Family 2:** First\_name, middle\_name, last\_name, profession.

[3 NF\family\_2.xlsx](3%20NF/family_2.xlsx)

**Family 3:** First\_name,middle\_name, last\_name, house\_no\_or\_name, area, city, state.

[3 NF\family\_3.xlsx](3%20NF/family_3.xlsx)

Entity 3: **F\_members** [3 NF\F\_members.xlsx](file:///C:\Users\Kriti%20Singhal\Desktop\dbms%20proj\3%20NF\F_members.xlsx)

Since there is no transitive dependency among any of the attributes, thus it is already in 3rd NF. Hence no further decomposition is required.

Entity 4: **Corporate**

In this relation, following transitive functional dependency exists-:

1. *company\_id* 🡪 *company\_name* 🡪 *factory\_name, area, city, state.*

Hence, we need to decompose the relation employee into following 2 relations:

**Corporate 1:** company\_name, factory\_name, area, city, state

[3 NF\Corporate 1.xlsx](3%20NF/Corporate%201.xlsx)

**Corporate 2:** company\_id, company\_name

[3 NF\Corporate 2.xlsx](3%20NF/Corporate%202.xlsx)

Entity 5: **corp\_members**

Since there is no transitive dependency among any of the attributes, thus it is already in 3rd NF. Hence no further decomposition is required.

[3 NF\corp\_members 1.xlsx](3%20NF/corp_members%201.xlsx)

[3 NF\corp\_members 2.xlsx](3%20NF/corp_members%202.xlsx)

Entity 6: **Food**

Since there is no transitive dependency among any of the attributes, thus it is already in 3rd NF. Hence no further decomposition is required.

[3 NF\Food Relation.xlsx](3%20NF/Food%20Relation.xlsx)

Entity 7: **Bill**

In this relation, following transitive functional dependency exists-:

1. *Bill\_no 🡪 food\_charges 🡪 vat*
2. *Bill\_no 🡪 room\_charges 🡪 lux\_tax*
3. *Bill\_no 🡪 room\_charges, food\_charges 🡪 ser\_tax*
4. *Bill\_no 🡪 food\_charges, room\_charges, vat, lux\_tax, ser\_tax 🡪 grand\_total*

Hence, we need to decompose the relation employee into following 5 relations:

**Bill 1:** bill\_no. bill\_date, room\_charges, food\_charges

[3 NF\BILL\_1.xlsx](3%20NF/BILL_1.xlsx)

**Bill2:** food\_charges, vat

[3 NF\bill\_2.xlsx](3%20NF/bill_2.xlsx)

**Bill 3:** room\_charges, lux\_tax

[3 NF\bill\_3.xlsx](3%20NF/bill_3.xlsx)

**Bill 4:** food\_charges, room\_charges, ser\_tax

[3 NF\bill\_4.xlsx](3%20NF/bill_4.xlsx)

**Bill 5:** food\_charges, room\_charges, ser\_tax, vat, lux\_tax, grand\_total

[3 NF\Bill\_5.xlsx](3%20NF/Bill_5.xlsx)

Entity 8: **Offers**[3 NF\Offers Relation.xlsx](file:///C:\Users\Kriti%20Singhal\Desktop\dbms%20proj\3%20NF\Offers%20Relation.xlsx)

Since there is no transitive dependency among any of the attributes, thus it is already in 3rd NF. Hence no further decomposition is required.

Entity 9: **Room**

In this relation, following transitive functional dependency exists-:

1. *Room\_no 🡪 room\_type 🡪 rate*

Hence, we need to decompose the relation employee into following 2 relations:

**Room 1:** room\_no, room\_type, status

[3 NF\Room Relation\_1.xlsx](3%20NF/Room%20Relation_1.xlsx)

**Room 2:** room\_type, rate.

[3 NF\room relation\_2.xlsx](3%20NF/room%20relation_2.xlsx)

Entity 10: **Employee**

In this relation, following transitive functional dependency exists-

1. *Emp\_id 🡪 emp\_mob\_no🡪 First\_name, middle\_name, last\_name*

(b) Emp\_id 🡪*Post 🡪 facility\_type*

Hence, we need to decompose the relation employee into following 3 relations:

**Employee\_1:** emp\_mb\_no, first\_name, middle\_name, last\_name

[3 NF\employee\_1.xlsx](3%20NF/employee_1.xlsx)

**Employe\_2:** emp\_id, emp\_mb\_no, post

[3 NF\employee\_2.xlsx](3%20NF/employee_2.xlsx)

**Employee 3:** post, facility\_type

[3 NF\employee\_3.xlsx](3%20NF/employee_3.xlsx)

Entity 11: **Facility**[3 NF\facility.xlsx](file:///C:\Users\Kriti%20Singhal\Desktop\dbms%20proj\3%20NF\facility.xlsx)

Since there is no transitive dependency among any of the attributes, thus it is already in 3rd NF. Hence no further decomposition is required.

**Relationships:**

Relationship 1: **Alloted**<Guest is alloted room.>

Since there is no transitive dependency among any of the attributes, thus it is already in 3rd NF. Hence no further decomposition is required.

[3 NF\guest\_alloted\_room.xlsx](3%20NF/guest_alloted_room.xlsx)

Relationship 2: **Orders**<Guest orders food.>

Since there is no transitive dependency among any of the attributes, thus it is already in 3rd NF. Hence no further decomposition is required.

[3 NF\guest\_orders\_food.xlsx](3%20NF/guest_orders_food.xlsx)

Relationship 3: **Pays <**Guest pays bill.>

Since there is no transitive dependency among any of the attributes, thus it is already in 3rd NF. Hence no further decomposition is required.

[3 NF\guest\_pays\_bill.xlsx](3%20NF/guest_pays_bill.xlsx)

Relationship 4: **Uses**<Guest uses facilities.>

Since there is no transitive dependency among any of the attributes, thus it is already in 3rd NF. Hence no further decomposition is required.

[3 NF\guest\_uses\_facilities.xlsx](3%20NF/guest_uses_facilities.xlsx)

**BOYCE CODD NORMAL FORM**

Boyce-Codd Normal Form (BCNF) is an extension of Third Normal Form on strict terms. BCNF states that −

* For any non-trivial functional dependency, X → A, X must be a super-key.

**Entities:**

Entity 1: **Guest**[bcnf\GUEST.xlsx](file:///C:\Users\Kriti%20Singhal\Desktop\dbms%20proj\bcnf\GUEST.xlsx)

In this relation the non-trivial functional dependency is-

*Reservation\_folio\_id 🡪 contact\_no, email\_id, no\_of\_members, arr\_frm, prcdng\_to, prps\_of\_visit, package\_type.*

Here reservation\_folio\_id is a super key, thus it is already in BCNF. Hence no further decomposition is required.

Entity 2: **Family**

In this relation the non-trivial functional dependencies are-

1. *Head\_id 🡪 first\_name, middle\_name, last\_name, no\_of\_children, no\_of\_adults, gender, age.*

Here head\_id is a super key, thus it is already in BCNF. Hence no further decomposition is required.

[bcnf\family\_1.xlsx](bcnf/family_1.xlsx)

1. *First\_name, middle\_name, last\_name 🡪 profession*

Here (First\_name, middle\_name, last\_name) are acting as a super key, thus it is already in BCNF. Hence no further decomposition is required.

[bcnf\family\_2.xlsx](bcnf/family_2.xlsx)

1. *First\_name, middle\_name, last\_name 🡪 house\_no\_or\_name, area, city, state*

Here (First\_name, middle\_name, last\_name) are acting as a super key, thus it is already in BCNF. Hence no further decomposition is required.

[bcnf\family\_3.xlsx](bcnf/family_3.xlsx)

Entity 3: **F\_members**[bcnf\F\_members.xlsx](file:///C:\Users\Kriti%20Singhal\Desktop\dbms%20proj\bcnf\F_members.xlsx)

In this relation the non-trivial functional dependencies are-

1. *Reservation\_folio\_id, first\_name, middle\_name, last\_name 🡪 age, gender*

Here (Reservation\_folio\_id, first\_name, middle\_name, last\_name) are acting as a super key, thus it is already in BCNF. Hence no further decomposition is required.

Entity 4: **Corporate**

In this relation the non-trivial functional dependencies are-

1. *Company\_name 🡪 factory\_name, area, city, state*

Here company\_name is acting as a super key, thus it is already in BCNF. Hence no further decomposition is required.

[bcnf\Corporate 1.xlsx](bcnf/Corporate%201.xlsx)

1. *Company\_id 🡪 company\_name*

Here company\_id is acting as a super key, thus it is already in BCNF. Hence no further decomposition is required.

[bcnf\Corporate 2.xlsx](bcnf/Corporate%202.xlsx)

Entity 5: **corp\_members**

In this relation the non-trivial functional dependencies are-

1. *Reservation\_folio\_id, contact\_no 🡪 first\_name, middle\_name, last\_name*

Here (Reservation\_folio\_id, contact\_no) are acting as a super key, thus it is already in BCNF. Hence no further decomposition is required.

[bcnf\corp\_members 1.xlsx](bcnf/corp_members%201.xlsx)

1. *first\_name, middle\_name, last\_name 🡪 age, gender, designation*

Here (first\_name, middle\_name, last\_name) are acting as a super key, thus it is already in BCNF. Hence no further decomposition is required.

[bcnf\corp\_members 2.xlsx](bcnf/corp_members%202.xlsx)

Entity 6: **Food**[bcnf\Food Relation.xlsx](file:///C:\Users\Kriti%20Singhal\Desktop\dbms%20proj\bcnf\Food%20Relation.xlsx)

In this relation the non-trivial functional dependency is-

1. *Pdct\_name 🡪 pdct\_price, food\_type*

Here pdct\_name is acting as a super key, thus it is already in BCNF. Hence no further decomposition is required.

Entity 7: **Bill**

In this relation the non-trivial functional dependencies are-

1. *Bill\_no 🡪 bill\_date, food\_charges,room\_charges*

Here bill\_no is acting as a super key, thus it is already in BCNF. Hence no further decomposition is required.

[bcnf\BILL\_1.xlsx](bcnf/BILL_1.xlsx)

1. *Food\_charges 🡪 vat*

Here food\_charges is acting as a super key, thus it is already in BCNF. Hence no further decomposition is required.

[bcnf\bill\_2.xlsx](bcnf/bill_2.xlsx)

1. *Room\_charges 🡪 lux\_tax*

Here room\_charges is acting as a super key, thus it is already in BCNF. Hence no further decomposition is required.

[bcnf\bill\_3.xlsx](bcnf/bill_3.xlsx)

1. *Room\_charges, food\_charges 🡪 ser\_tax*

Here(food\_charges, room\_charges) are acting as a super key, thus it is already in BCNF. Hence no further decomposition is required.

[bcnf\bill\_4.xlsx](bcnf/bill_4.xlsx)

1. *Room\_charges, food\_charges, ser\_tax, vat, lux\_tax 🡪 grand\_total*

Here (Room\_charges, food\_charges, ser\_tax, vat, lux\_tax) are acting as a super key, thus it is already in BCNF. Hence no further decomposition is required.

[bcnf\Bill\_5.xlsx](bcnf/Bill_5.xlsx)

Entity 8: **Offers**[bcnf\Offers Relation.xlsx](file:///C:\Users\Kriti%20Singhal\Desktop\dbms%20proj\bcnf\Offers%20Relation.xlsx)

In this relation the non-trivial functional dependency is-

1. *Package\_type 🡪 description*

Here package\_type is acting as a super key, thus it is already in BCNF. Hence no further decomposition is required.

Entity 9: **Room**

In this relation the non-trivial functional dependencies are-

1. *Room\_no 🡪 room\_type, status*

Here room\_no is acting as a super key, thus it is already in BCNF. Hence no further decomposition is required.

[bcnf\Room Relation\_1.xlsx](bcnf/Room%20Relation_1.xlsx)

1. *Room\_type 🡪 rate*

Here room\_type is acting as a super key, thus it is already in BCNF. Hence no further decomposition is required.

[bcnf\room relation\_2.xlsx](bcnf/room%20relation_2.xlsx)

Entity 10: **Employee**

In this relation the non-trivial functional dependencies are-

1. *Emp\_mb\_no 🡪 first\_name, middle\_name, last\_name.*

Here emp\_mb\_no is acting as a super key, thus it is already in BCNF. Hence no further decomposition is required.

[bcnf\employee\_1.xlsx](bcnf/employee_1.xlsx)

1. *Emp\_id 🡪 emp\_mb\_no, post*

Here emp\_id is acting as a super key, thus it is already in BCNF. Hence no further decomposition is required.

[bcnf\employee\_2.xlsx](bcnf/employee_2.xlsx)

1. *Post🡪 facility\_type*

Here post is acting as a super key, thus it is already in BCNF. Hence no further decomposition is required.

[bcnf\employee\_3.xlsx](bcnf/employee_3.xlsx)

Entity 11: **Facility**[**bcnf\facility.xlsx**](bcnf/facility.xlsx)

In this relation the non-trivial functional dependency is-

1. *Facility\_type 🡪 facility\_rate*

Here facility\_type is acting as a super key, thus it is already in BCNF. Hence no further decomposition is required.

**Relationships**

Relationship 1: **Alloted<**Guest is alloted room.>

In this relation the non-trivial functional dependency is-

1. *Reservation\_folio\_id, room\_no 🡪 cin\_date, cin\_time, cout\_date, cout\_time*

Here (Reservation\_folio\_id, room\_no) are acting as a super key, thus it is already in BCNF. Hence no further decomposition is required.

[bcnf\guest\_alloted\_room.xlsx](bcnf/guest_alloted_room.xlsx)

Relationship 2: **Orders<**Guest orders food.>

In this relation the non-trivial functional dependency is-

1. *Reservation\_folio\_id, pdct\_name 🡪 order\_no, date, time, qty.*

Here (Reservation\_folio\_id, pdct\_name) are acting as a super key, thus it is already in BCNF. Hence no further decomposition is required.

[bcnf\guest\_orders\_food.xlsx](bcnf/guest_orders_food.xlsx)

Relationship 3: **Pays<**Guest pays bill.>

In this relation the non-trivial functional dependency is-

1. *Reservation\_folio\_id, bill\_no 🡪 payment\_method*

Here (Reservation\_folio\_id, bill\_no) are acting as a super key, thus it is already in BCNF. Hence no further decomposition is required.

[bcnf\guest\_pays\_bill.xlsx](bcnf/guest_pays_bill.xlsx)

Relationship 4: **Uses**<Guest uses facilities.>

In this relation the non-trivial functional dependency is-

1. *Reservation\_folio\_id 🡪 facility\_type*

Here Reservation\_folio\_id is acting as a super key, thus it is already in BCNF. Hence no further decomposition is required.

[bcnf\guest\_uses\_facilities.xlsx](bcnf/guest_uses_facilities.xlsx)

STAKEHOLDERS AND THEIR RESPECTIVE QUERIES

**RECEPTIONIST**

Receptionist is the person who directly communicates with the customer. The possible queries of the customers to be answered are as follows:

**QUERIES:**

1. *Vacancy of all rooms:*

SELECT \*

FROM room\_relation\_1

WHERE STATUS = ’vacant’

1. *Vacancy of particular room type:*

SELECT \*

FROM room\_relation\_1

WHERE status='vacant'

AND room\_type='single nonac'

1. *Count of vacant rooms:*

SELECT count(room\_no)

FROM room\_relation\_1

WHERE status='vacant'

1. *Room charges:*

SELECT \*

FROM room\_relation\_2

1. *Details of all the facilities:*

SELECT \*

FROM facility

1. *Offers’ details:*

SELECT \*

FROM offers

1. *Employee details:*

SELECT first\_name, middle\_name, last\_name, employee1.emp\_mb\_no

FROM employee1, employee2

WHERE employee1.emp\_mb\_no=employee2.emp\_mb\_no and post='manager'

1. *Updating check out date and time:*

UPDATE guest\_alloted\_room

SET cout\_date = '13.02.2016', cout\_time = '2:15pm' WHERE reservation\_folio\_id = 123457

1. *Updating facilities:*

INSERT into facility(facility\_type, facility\_rate) VALUES ('beautician', 1000)

1. *Information of a particular guest:*

SELECT \*

FROM family\_1

JOIN family\_2

ON

family\_1.reservation\_folio\_id=family\_2.reservation\_folio\_id

JOIN family\_3

ON family\_1.reservation\_folio\_id=family\_3.reservation\_folio\_id

JOIN guest

ON family\_1.reservation\_folio\_id=guest.resrevation\_folio\_id

1. *Generation of a complete bill for the guest:*

SELECT

bill\_1.bill\_no,

bill\_1.bill\_date,

guest\_alloted\_room.room\_no,

guest\_alloted\_room.cin\_date,

guest\_alloted\_room.cout\_date,

bill\_1.food\_charges,

bill\_1.room\_charges,

bill\_1.vat,

bill\_1.ser\_tax,

bill\_1.lux\_tax,

bill\_1.grand\_total,

guest\_pays\_bill.payment\_method,

guest\_pays\_bill.reservation\_folio\_id,

family\_1.first\_name,

family\_1.middle\_name,

family\_1.last\_name

FROM

guest\_pays\_bill

JOIN

bill\_1

ON guest\_pays\_bill.bill\_no = bill\_1.bill\_no

JOIN guest\_alloted\_room

ON

guest\_pays\_bill.reservation\_folio\_id = guest\_alloted\_room.reservation\_folio\_id

JOIN family\_1

ON

guest\_pays\_bill.reservation\_folio\_id = family\_1.reservation\_folio\_id

WHERE

family\_1.first\_name = 'harsh'

12. *Generating Separate food bill:*

a. *Food items list and price:*

SELECT

guest\_orders\_food.reservation\_folio\_id,

guest\_orders\_food.pdct\_name,

guest\_orders\_food.order\_no,

guest\_orders\_food.date,

guest\_orders\_food.time,

guest\_orders\_food.qty,

food\_relation.pdct\_price

FROM

guest\_orders\_food,

food\_relation

WHERE

guest\_orders\_food.pdct\_name = food\_relation.pdct\_name AND guest\_orders\_food.reservation\_folio\_id = 1267

b. *Total food charges:*

SELECT SUM

( food\_relation.pdct\_price \* guest\_orders\_food.qty )

FROM

guest\_orders\_food,

food\_relation

WHERE

guest\_orders\_food.pdct\_name = food\_relation.pdct\_name AND guest\_orders\_food.reservation\_folio\_id = 1267

13. *Generating separate facility bill:*

SELECT

guest\_uses\_facility.reservation\_folio\_id,

guest\_uses\_facility.facility\_type,

facility.facility\_rate

FROM

guest\_uses\_facility,

facility

WHERE

guest\_uses\_facility.facility\_type = facility.facility\_type AND guest\_uses\_facility.reservation\_folio\_id = 123456

RESTAURANT MANAGER:

There may be some guests who only seek restaurant facility. In such cases, a restaurant manager’s responsibility is to generate and maintain restaurant bills.

**QUERIES:**

1. *Bill generation:*

SELECT SUM(pdct\_price)

FROM food\_relation

WHERE pdct\_name

IN('Bangda Masala','Chicken Biryani')

HOTEL OWNER

Hotel owners are active stakeholders whose key motive is to keep an eye on the performance of the hotel.

**QUERIES:**

1. *No of guest check outs on a particular day:*

SELECT count(cout\_date)

FROM guest\_alloted\_room

WHERE cout\_date='18.04.2014'

1. *No of guests currently residing in hotel:*

SELECT count(\*)

FROM guest\_alloted\_room

WHERE cout\_date IS NULL

1. *Total bills generated on a particular day:*

SELECT count(cout\_date)

FROM guest\_alloted\_room

WHERE cout\_date='18.04.2014'