Project for Database Design:

Phase II. Relational Schema

Kinjal Basu
kxb170730@utdallas.edu

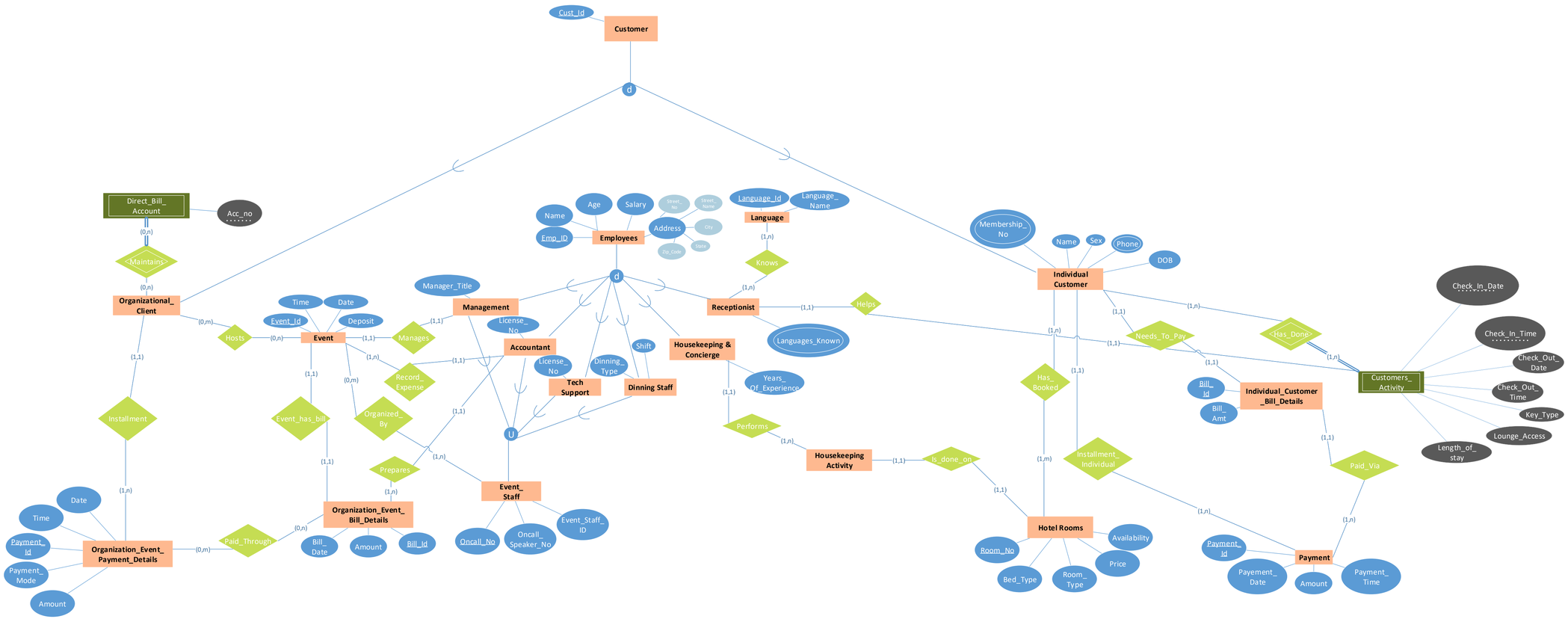
Piyush Mahatkar
pkm170230@utdallas.edu

1. Modified EER diagram:

We have done the following changes in the EER diagram:

- In Customer Activity weak Entity, we have added the receptionist attribute and time attribute
- Remove Dinning staff specialization and added a new attribute of type.
- Create a new relation between event and bill

The diagram is on the next Page.

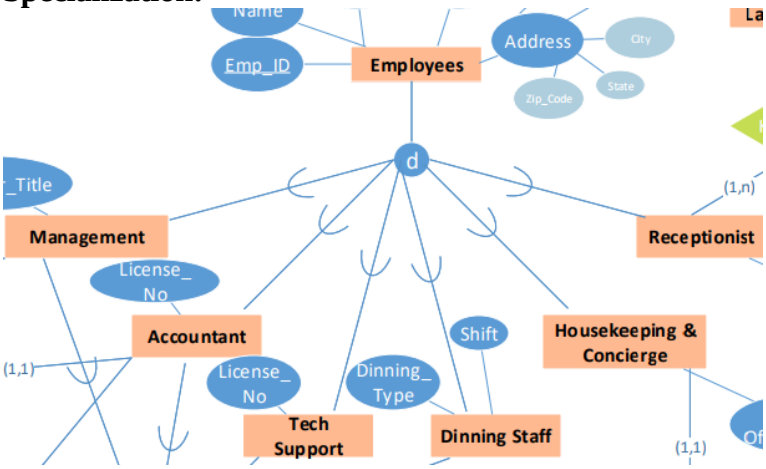

Cust_Id
Customer
d
Direct_Bill_ Account
Acc_no
(0,n)
<Maintains>
(0,n)
Organizational_ Client
(0,m)
Hosts
(0,n)
Event
(1,1)
Manages
(1,1)
Time
Date
Event_Id
Deposit
(1,1)
(1,n)
Record_ Expense
(1,1)
(1,1)
Installment
(1,n)
Event_has_bill
(1,1)
(0,m)
Organized_ By
(1,1)
(1,n)
Prepares
(1,n)
Date
Time
Payment_ Id
Payment_ Mode
Amount
Organization_Event_ Payment_Details
(0,m)
Paid_Through
(0,n)
Organization_Event_ Bill_Details
Bill_ Date
Amount
Bill_Id
Event_ Staff
Oncall_No
Oncall_ Speaker_No
Event_Staff_ ID
U
Manager_Title
Management
(1,1)
License_ No
Accountant
License_ No
Tech Support
Dinning_ Type
Shift
Dinning Staff
Name
Age
Salary
Address
Street_ No
Street_ Name
City
State
Zip_Code
Emp_ID
Employees
Language_Id
Language_ Name
Language
(1,n)
Knows
(1,n)
d
Receptionist
(1,1)
Helps
Languages_Known
Housekeeping & Concierge
Years_ Of_Experience
(1,1)
Performs
(1,n)
Housekeeping Activity
(1,1)
Is_done_on
(1,1)
Membership_ No
Name
Sex
Phone
DOB
Individual Customer
(1,1)
(1,n)
(1,n)
Needs_To_Pay
(1,1)
Has_ Booked
(1,m)
(1,1)
Installment_ Individual
(1,n)
Hotel Rooms
Room_No
Bed_Type
Room_ Type
Availability
Price
Bill_ Id
Bill_ Amt
Individual_Customer _Bill_Details
(1,1)
Paid_Via
(1,n)
Payment_ Id
Payement_ Date
Amount
Payment_ Time
Payment
<Has_Done>
(1,n)
Customers_ Activity
Check_In_Date
Check_In_Time
Check_Out_ Date
Check_Out_ Time
Key_Type
Lounge_Access
Length_of_ stay

2. Mapping Relational Schemas

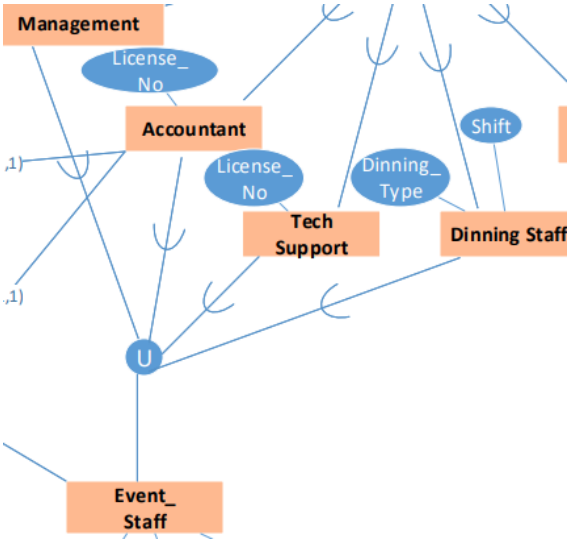
2.1 For each regular entity type in the EER Diagram, we created a relation that includes all the simple attributes. We Included only the simple component attributes of a composite attribute. We chose one of the key attributes as the primary key.
In our EER Diagram, Organization_Event_ Payment_Details , Event, Organization_Event_ Bill_Details, language, Hotel_rooms, Payment, Individual_Customer _Bill_Details , and Language are the Regular entities.

For Specialization we have used the strategy of using all the attributes for sub classes and super classes. Our EER Diagram has specializations and unions. Employee Entity is specialized to accountant, management, Tech support, housekeeping and conceige, Dining staff and Receptionist. Customer is specialized to Organizational client and individual customer. We have also created a Event_staff entity as a Union for Management, Tech_Support, Dining staff and accountant.

Specialization:

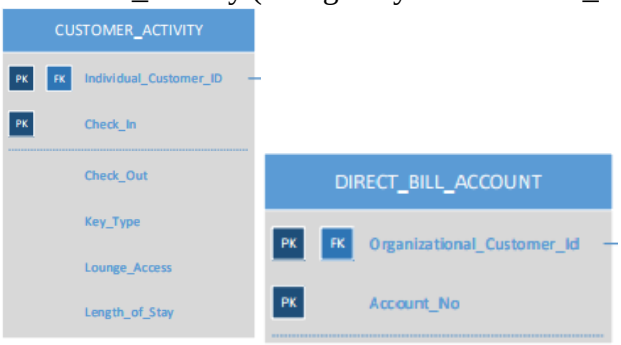


Union:



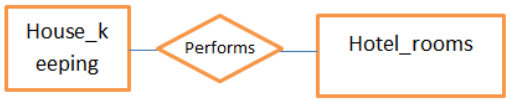
2.2 Mapping of Weak Entity Types:

For each weak entity type in the ER schema with owner entity type, we create a relation and include all simple attributes as attributes. In addition, include as foreign key attributes, the primary key attribute(s) of the relation that correspond to the owner entity type; this takes care of mapping the identifying relationship type. The primary key here is the combination of the primary key(s) of the owner(s) and the partial key of the weak entity type.
We have Direct_bill_account (foreign key with organizational_customer_id) and Customers_activity (foreign key as Individual_customer_Id) as week entities.



2.3 Mapping of Binary 1:1 Relationship Types

As an example, Mapping to Binary 1:1 Relationship between Housekeeping and Hotel_Rooms.

Relation	Mapping Method
	Since it is a one to one relationship, we use the foreign key approach. Since on Hotel_Rooms side, the relationship is total participation, we include housekeeping_id as a foreign key in Hotel_Rooms

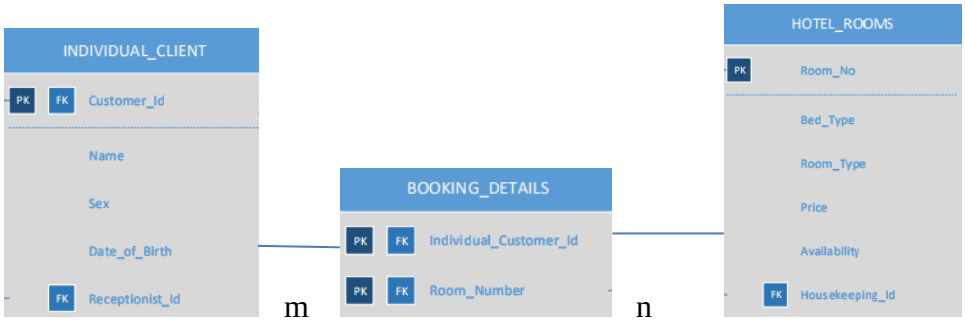
2.4 Mapping of Binary 1:N Relationship Types

Similar mapping of 1:N occurs in Accountant and Event.

Relation	Mapping Method
	Since it is a one to many relationship, The N-side of this relationship type is Event. Thus we include the primary key Accountant_Id of the relation Accountant as foreign key in relation Event.

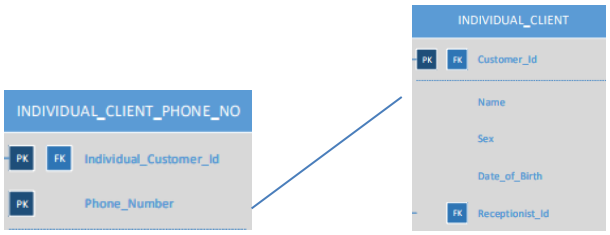
2.5 Mapping of Binary M:N Relationship Types

Here there is a M:N relationship between Individual_Client and Hotel_Rooms. This mapping is done using a separate relation table which has a foreign key from both the entities. Hence this new table is named as Booking_Details.



2.6 Mapping of Multi-valued Attributes

We are creating a new entity for multivalued attributes, with primary key of referring relation as the foreign key in the new relation.

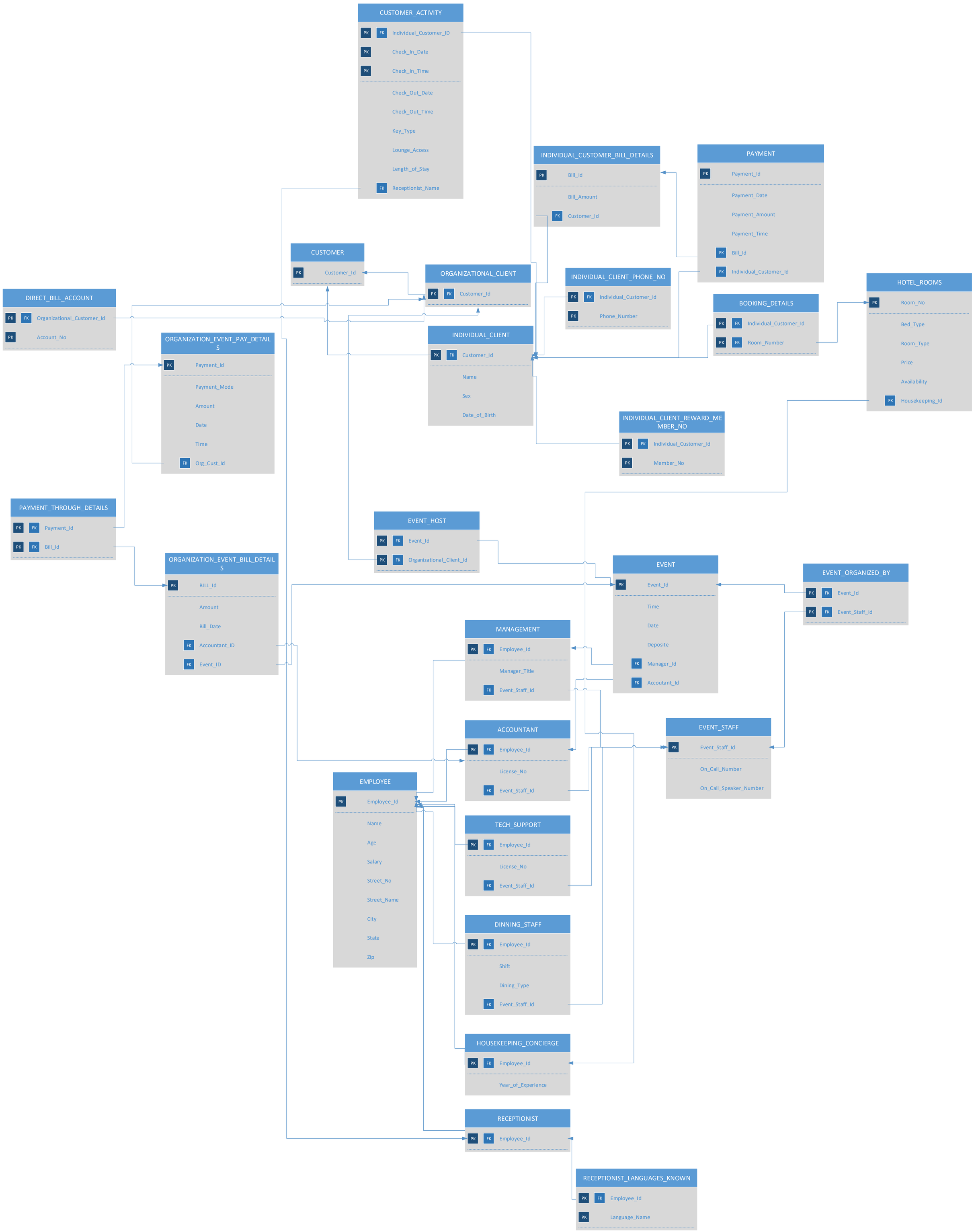


2.7 Mapping of N-ary Relationship Types

There are no n-ary relationships used in the design.

2.8 Final Relational Schema.

The Diagram is on the next page.


CUSTOMER_ACTIVITY
PK FK Individual_Customer_ID
PK Check_In_Date
PK Check_In_Time
Check_Out_Date
Check_Out_Time
Key_Type
Lounge_Access
Length_of_Stay
FK Receptionist_Name
INDIVIDUAL_CUSTOMER_BILL_DETAILS
PK Bill_Id
Bill_Amount
FK Customer_Id
PAYMENT
PK Payment_Id
Payment_Date
Payment_Amount
Payment_Time
FK Bill_Id
FK Individual_Customer_Id
CUSTOMER
PK Customer_Id
ORGANIZATIONAL_CLIENT
PK FK Customer_Id
INDIVIDUAL_CLIENT_PHONE_NO
PK FK Individual_Customer_Id
PK Phone_Number
HOTEL_ROOMS
PK Room_No
Bed_Type
Room_Type
Price
Availability
FK Housekeeping_Id
DIRECT_BILL_ACCOUNT
PK FK Organizational_Customer_Id
PK Account_No
ORGANIZATION_EVENT_PAY_DETAILS
PK Payment_Id
Payment_Mode
Amount
Date
TIme
FK Org_Cust_Id
INDIVIDUAL_CLIENT
PK FK Customer_Id
Name
Sex
Date_of_Birth
BOOKING_DETAILS
PK FK Individual_Customer_Id
PK FK Room_Number
INDIVIDUAL_CLIENT_REWARD_MEMBER_NO
PK FK Individual_Customer_Id
PK Member_No
PAYMENT_THROUGH_DETAILS
PK FK Payment_Id
PK FK Bill_Id
EVENT_HOST
PK FK Event_Id
PK FK Organizational_Client_Id
ORGANIZATION_EVENT_BILL_DETAILS
PK BILL_Id
Amount
Bill_Date
FK Accountant_ID
FK Event_ID
EVENT
PK Event_Id
Time
Date
Deposite
FK Manager_Id
FK Accoutant_Id
EVENT_ORGANIZED_BY
PK FK Event_Id
PK FK Event_Staff_Id
MANAGEMENT
PK FK Employee_Id
Manager_Title
FK Event_Staff_Id
ACCOUNTANT
PK FK Employee_Id
License_No
FK Event_Staff_Id
EVENT_STAFF
PK Event_Staff_Id
On_Call_Number
On_Call_Speaker_Number
EMPLOYEE
PK Employee_Id
Name
Age
Salary
Street_No
Street_Name
City
State
Zip
TECH_SUPPORT
PK FK Employee_Id
License_No
FK Event_Staff_Id
DINNING_STAFF
PK FK Employee_Id
Shift
Dining_Type
FK Event_Staff_Id
HOUSEKEEPING_CONCIERGE
PK FK Employee_Id
Year_of_Experience
RECEPTIONIST
PK FK Employee_Id
RECEPTIONIST_LANGUAGES_KNOWN
PK FK Employee_Id
PK Language_Name

3. Documentation for schemas

3.1 Explanation for format design

Rules and assumptions:

- Employee_ID is a 9-Digit Number.
- Years_of _experience for housekeeping_staff is range(0-50)
- Individual_Customer id is a 6 digit number.
- Phone number for customer is a 10 digit number of format (xxx)xxx-xxxx
- Date_of_Birth format is MM/DD/YYYY
- Room number is a 4 digit number.
- check-in date ('MM/DD/YYYY') + time ('HH-MM-SS')
- key type has one of the two values (card key or digital key),
- lounge access has one of the two values (Yes/No)
- organization client is uniquely identified by an ID (6-digit number)
- Every event staff member is assigned on-call number (4-digit number)
- Each event has an event ID (4-digit number).
- Each bill has a unique ID (6-digit number), with date issued format ('MM/DD/YYYY') and total amount in numbers.

3.2 Format for Every Relation

Customer Activity:	Data Type
Individual_Customer_ID	Integer (6 digit)
Check_In	MM/DD/YYYY
Check_Out	MM/DD/YYYY
Key_Type	Boolean
Lounge_Access	(Card(0)/Digital(1))
Length_of_Stay	Boolean(Yes/No)
	Integer

CUSTOMER:	Data Type
Customer_Id	Integer (6

	digit)

ORGANIZATION_EVENT_PAY_DETAILS	Data Type
Payment_Id	Integer
Payment_Mode	String <chars(20)>
Amount	Integer
Date	MM/DD/YYYY
Time	HH:MM:SS
Org_Cust_Id	Integer (6 digit)

DIRECT_BILL_ACCOUNT	Data Type
Organizational_Customer_Id	Integer (6 digit)
Account_No	Integer

PAYMENT_THROUGH_DETAILS	Data Type
Payment_Id	Integer
Bill_Id	Integer(6 Digit)

ORGANIZATION_EVENT_BILL_DETAILs	Data Type
BILL_Id	Integer (6 digit)
Amount	Integer
Bill_Date	MM/DD/YYYY
Accountant_ID	Integer
Event_ID	Integer (4 digit)

ORGANIZATIONAL_CLIENT	Data Type
Customer_Id	Integer (6 digit)

INDIVIDUAL_CLIENT	Data Type
Customer_Id	Integer (6 digit)

Name	String (20 Chars)
Sex	M/F
Date_of_Birth	MM/DD/YYYY
Receptionist_Id	Integer

EVENT_HOST	Data Type
Event_Id	Integer (4 digit)
Organizational_Client_Id	Integer (6 digit)

MANAGEMENT	Data Type
Employee_Id	Integer (9 digit)
Manager_Title	String (20 chars)
Event_Staff_Id	Integer

ACCOUNTANT	Data type
Employee_Id	Integer (9 digit)
License_No	Integer
Event_Staff_Id	Integer

TECH_SUPPORT	Data Type
Employee_Id	Integer (9 digit)
License_No	Integer
Event_Staff_Id	Integer

DINNING_STAFF	Data Type
Employee_Id	Integer (9 digit)
Shift	string
Dining_Type	string
Event_Staff_Id	Integer

HOUSEKEEPING_CONCIERGE	Data Type
Employee_Id	Integer (9 digit)
Year_of_Experience	Integer (range (0-50))

RECEPTIONIST	Data Type
Employee_Id	Integer (9 digit)
Year_of_Experience	Integer (range (0-50))

EMPLOYEE	Data Type
Employee_Id	Integer (9 digit)
Name	String
Age	Integer
Salary	Integer
Street_No	Integer
Street_Name	String
City	String
State	String
Zip	Integer

EVENT	Data Type
Event_Id	Integer (4 digit)
Time	HH:MM:SS
Date	MM/DD/YYYY
Deposite	Integer
Manager_Id	Integer
Accoutant_Id	Integer

EVENT_STAFF	Data Type
Event_Staff_Id	Integer
On_Call_Number	Integer (4 digit)
On_Call_Speaker_Number	Integer

EVENT_ORGANIZED_BY	Data Type
Event_Id	Integer (4 digit)
Event_Staff_Id	Integer

INDIVIDUAL_CUSTOMER_BILL_DETAILS	Data Type
Bill_Id	Integer (6 digit)
Bill_Amount	Integer

Customer_Id	Integer (6 digit)

INDIVIDUAL_CLIENT_PHONE_NO	Data Type
Individual_Customer_Id	Integer (6 digit)
Phone_Number	Integer (10 digit)

PAYMENT	Data Type
Payment_Id	Integer
Payment_Date	MM/DD/YYYY
Payment_Time	HH:MM:SS
Bill_Id	Integer (6 digit)
Individual_Customer_Id	Integer (6 digit)

BOOKING_DETAILS	Data Type
Individual_Customer_Id	Integer (6 digit)
Room_Number	Integer (4 digit)

INDIVIDUAL_CLIENT_REWARD_MEMBER_NO	Data Type
Individual_Customer_Id	Integer (6 digit)
Member_No	Integer

HOTEL_ROOMS	Data Type
Room_No	Integer (4 digit)
Bed_Type	String
Room_Type	String
Price	Integer
Availability	Boolean
Housekeeping_Id	Integer

ACCOUNTANT	Data Type
Employee_Id	Integer (9 digit)
License_No	Integer
Event_Staff_Id	Integer

3.3. Implement the Database:

Following are the queries and the table snippet from sql developer tool.

1.

```
CREATE TABLE EMPLOYEE (
   Employee_Id NUMBER(9) NOT NULL PRIMARY KEY,
   Manager_Titlte varchar(100),
   Age NUMBER(3) CHECK (Age >=0),
   Salary NUMBER(20) CHECK (Salary >=0),
   Street_no NUMBER(10),
   Street_name varchar(30),
   City varchar(20),
   State varchar(20),
   Zip NUMBER(32)
);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	EMPLOYEE_ID	NUMBER(9,0)	No	(null)	1	(null)
2	EMP_NAME	VARCHAR2(100 BYTE)	Yes	(null)	2	(null)
3	AGE	NUMBER(3,0)	Yes	(null)	3	(null)
4	SALARY	NUMBER(20,0)	Yes	(null)	4	(null)
5	STREET_NO	NUMBER(10,0)	Yes	(null)	5	(null)
6	STREET_NAME	VARCHAR2(30 BYTE)	Yes	(null)	6	(null)
7	CITY	VARCHAR2(20 BYTE)	Yes	(null)	7	(null)
8	STATE	VARCHAR2(20 BYTE)	Yes	(null)	8	(null)
9	ZIP	NUMBER(32,0)	Yes	(null)	9	(null)

2.

```
CREATE TABLE RECEPTIONIST (
   Employee_Id NUMBER(9) NOT NULL PRIMARY KEY,
   FOREIGN KEY (Employee_Id) REFERENCES EMPLOYEE(Employee_Id)
);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	EMPLOYEE_ID	NUMBER(9,0)	No	(null)	1	(null)

3.

```
CREATE TABLE HOUSEKEEPING_CONCIERGE (
   Employee_Id NUMBER(9) NOT NULL PRIMARY KEY,
```

```
   Years_of_Experience NUMBER(2) CHECK (Years_of_Experience >=0 AND Years_of_Experience<=50),
   FOREIGN KEY (Employee_Id) REFERENCES EMPLOYEE(Employee_Id)
);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	EMPLOYEE_ID	NUMBER(9,0)	No	(null)	1	(null)
2	YEARS_OF_EXPERIENCE	NUMBER(2,0)	Yes	(null)	2	(null)

4.

```
CREATE TABLE EVENT_STAFF (
   Event_Staff_Id NUMBER(9) NOT NULL PRIMARY KEY,
   On_Call_Number NUMBER(4) NOT NULL,
   On_Call_Speaker_Number NUMBER(20) NOT NULL
);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	EVENT_STAFF_ID	NUMBER(9,0)	No	(null)	1	(null)
2	ON_CALL_NUMBER	NUMBER(4,0)	No	(null)	2	(null)
3	ON_CALL_SPEAKER_NUMBER	NUMBER(20,0)	No	(null)	3	(null)

5.

```
CREATE TABLE TECH_SUPPORT (
   Employee_Id NUMBER(9) NOT NULL PRIMARY KEY,
   License_No NUMBER(20) NOT NULL,
   Event_Staff_Id NUMBER(9),
   FOREIGN KEY (Employee_Id) REFERENCES EMPLOYEE(Employee_Id),
   FOREIGN KEY (Event_Staff_Id) REFERENCES EVENT_STAFF(EVENT_STAFF_ID)
);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	EMPLOYEE_ID	NUMBER(9,0)	No	(null)	1	(null)
2	LICENSE_NO	NUMBER(20,0)	No	(null)	2	(null)
3	EVENT_STAFF_ID	NUMBER(9,0)	Yes	(null)	3	(null)

6.

```
CREATE TABLE ACCOUNTANT (
   Employee_Id NUMBER(9) NOT NULL PRIMARY KEY,
   License_No NUMBER(20) NOT NULL,
   Event_Staff_Id NUMBER(9),
   FOREIGN KEY (Employee_Id) REFERENCES EMPLOYEE(Employee_Id),
   FOREIGN KEY (Event_Staff_Id) REFERENCES EVENT_STAFF(EVENT_STAFF_ID)
);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	EMPLOYEE_ID	NUMBER(9,0)	No	(null)	1	(null)
2	LICENSE_NO	NUMBER(20,0)	No	(null)	2	(null)
3	EVENT_STAFF_ID	NUMBER(9,0)	Yes	(null)	3	(null)

7.

```
CREATE TABLE RECEPTIONIST_LANGUAGES_KNOWN (
   Employee_Id NUMBER(9) NOT NULL,
   Language_Name VARCHAR(30) NOT NULL,
   FOREIGN KEY (Employee_Id) REFERENCES RECEPTIONIST(Employee_Id),
   PRIMARY KEY (Employee_Id, Language_Name)
);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	EMPLOYEE_ID	NUMBER(9,0)	No	(null)	1	(null)
2	LANGUAGE_NAME	VARCHAR2(30 BYTE)	No	(null)	2	(null)

8.

```
CREATE TABLE DINING_STAFF (
   Employee_Id NUMBER(9) NOT NULL PRIMARY KEY,
   Shift VARCHAR(20) NOT NULL CHECK (Shift IN ('morning', 'afternoon', 'evening', 'night')),
   Dining_Type VARCHAR(20) NOT NULL,
   Event_Staff_Id NUMBER(9),
   FOREIGN KEY (Employee_Id) REFERENCES EMPLOYEE(Employee_Id),
   FOREIGN KEY (Event_Staff_Id) REFERENCES EVENT_STAFF(EVENT_STAFF_ID)
);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	EMPLOYEE_ID	NUMBER(9,0)	No	(null)	1	(null)
2	SHIFT	VARCHAR2(20 BYTE)	No	(null)	2	(null)
3	DINING_TYPE	VARCHAR2(20 BYTE)	No	(null)	3	(null)
4	EVENT_STAFF_ID	NUMBER(9,0)	Yes	(null)	4	(null)

9.

```
CREATE TABLE MANAGEMENT (
   Employee_Id NUMBER(9) NOT NULL PRIMARY KEY,
   Manager_Title VARCHAR(20) NOT NULL,
   Event_Staff_Id NUMBER(9),
   FOREIGN KEY (Employee_Id) REFERENCES EMPLOYEE(Employee_Id),
   FOREIGN KEY (Event_Staff_Id) REFERENCES EVENT_STAFF(EVENT_STAFF_ID)
);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	EMPLOYEE_ID	NUMBER(9,0)	No	(null)	1	(null)
2	MANAGER_TITLE	VARCHAR2(20 BYTE)	No	(null)	2	(null)
3	EVENT_STAFF_ID	NUMBER(9,0)	Yes	(null)	3	(null)

10.

```
CREATE TABLE EVENT(
   EVENT_ID NUMBER(4) NOT NULL PRIMARY KEY,
   EVENT_TIME TIMESTAMP NOT NULL,
   EVENT_DATE DATE NOT NULL,
   DEPOSIT NUMBER(20),
   MANAGER_ID NUMBER(9) NOT NULL,
   ACCOUNTANT_ID NUMBER(9)NOT NULL,
   FOREIGN KEY (MANAGER_ID) REFERENCES MANAGEMENT(EMPLOYEE_ID),
   FOREIGN KEY (ACCOUNTANT_ID) REFERENCES ACCOUNTANT(EMPLOYEE_ID)
);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	EVENT_ID	NUMBER(4,0)	No	(null)	1	(null)
2	EVENT_TIME	TIMESTAMP(6)	No	(null)	2	(null)
3	EVENT_DATE	DATE	No	(null)	3	(null)
4	DEPOSIT	NUMBER(20,0)	Yes	(null)	4	(null)
5	MANAGER_ID	NUMBER(9,0)	No	(null)	5	(null)
6	ACCOUNTANT_ID	NUMBER(9,0)	No	(null)	6	(null)

11.

```
CREATE TABLE EVENT_ORGANIZED_BY (
   EVENT_ID NUMBER(4) NOT NULL,
   EVENT_STAFF_ID NUMBER(9) NOT NULL,
   FOREIGN KEY (EVENT_ID) REFERENCES EVENT(EVENT_ID),
   FOREIGN KEY (EVENT_STAFF_ID) REFERENCES EVENT_STAFF(EVENT_STAFF_ID),
   PRIMARY KEY (EVENT_ID,EVENT_STAFF_ID)
);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	EVENT_ID	NUMBER(4,0)	No	(null)	1	(null)
2	EVENT_STAFF_ID	NUMBER(9,0)	No	(null)	2	(null)

12.

```
CREATE TABLE CUSTOMER(
   CUSTOMER_ID NUMBER(6) NOT NULL PRIMARY KEY
);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	CUSTOMER_ID	NUMBER(6,0)	No	(null)	1	(null)

13.

```
CREATE TABLE INDIVIDUAL_CLIENT(
   CUSTOMER_ID NUMBER(6) NOT NULL PRIMARY KEY,
   CUST_NAME VARCHAR(30) NOT NULL,
   SEX VARCHAR(10) NOT NULL,
   DOB DATE NOT NULL,
   FOREIGN KEY (CUSTOMER_ID) REFERENCES CUSTOMER(CUSTOMER_ID)
);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	CUSTOMER_ID	NUMBER(6,0)	No	(null)	1	(null)
2	CUST_NAME	VARCHAR2(30 BYTE)	No	(null)	2	(null)
3	SEX	VARCHAR2(10 BYTE)	No	(null)	3	(null)
4	DOB	DATE	No	(null)	4	(null)

14.

```
CREATE TABLE ORGANIZATIONAL_CLIENT(
   CUSTOMER_ID NUMBER(6) NOT NULL PRIMARY KEY,
   FOREIGN KEY (CUSTOMER_ID) REFERENCES CUSTOMER(CUSTOMER_ID)
);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	CUSTOMER_ID	NUMBER(6,0)	No	(null)	1	(null)

15.

```
CREATE TABLE CUSTOMER_ACTIVITY(
   INDIVIDUAL_CUSTOMER_ID NUMBER(6) NOT NULL,
   CHECK_IN_DATE DATE NOT NULL,
   CHECK_IN_TIME TIMESTAMP NOT NULL,
   CHECK_OUT_DATE DATE,
   CHECK_OUT_TIME TIMESTAMP,
   KEY_TYPE VARCHAR(10) NOT NULL CHECK(KEY_TYPE IN ('card key','digital key')),
   LOUNGE_ACCESS VARCHAR(5) CHECK(LOUNGE_ACCESS IN ('yes','no')) NOT NULL,
   LENGTH_OF_STAY NUMBER(2) CHECK(LENGTH_OF_STAY >=0),
   RECEPTIONIST_ID NUMBER(9) NOT NULL,
   PRIMARY KEY (INDIVIDUAL_CUSTOMER_ID, CHECK_IN_DATE, CHECK_IN_TIME),
   FOREIGN KEY (RECEPTIONIST_ID) REFERENCES RECEPTIONIST(EMPLOYEE_ID),
   FOREIGN KEY (INDIVIDUAL_CUSTOMER_ID) REFERENCES INDIVIDUAL_CLIENT(CUSTOMER_ID)

);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	INDIVIDUAL_CUSTOMER_ID	NUMBER(6,0)	No	(null)	1	(null)
2	CHECK_IN_DATE	DATE	No	(null)	2	(null)
3	CHECK_IN_TIME	TIMESTAMP(6)	No	(null)	3	(null)
4	CHECK_OUT_DATE	DATE	Yes	(null)	4	(null)
5	CHECK_OUT_TIME	TIMESTAMP(6)	Yes	(null)	5	(null)
6	KEY_TYPE	VARCHAR2(10 BYTE)	No	(null)	6	(null)
7	LOUNGE_ACCESS	VARCHAR2(5 BYTE)	No	(null)	7	(null)
8	LENGTH_OF_STAY	NUMBER(2,0)	Yes	(null)	8	(null)
9	RECEPTIONIST_ID	NUMBER(9,0)	No	(null)	9	(null)

16.

```
CREATE TABLE INDIVIDUAL_CUST_BILL_DETAILS(
   BILL_ID NUMBER(10) NOT NULL PRIMARY KEY,
   BILL_AMOUNT NUMBER(6) NOT NULL CHECK(BILL_AMOUNT >= 0),
   INDIVIDUAL_CUSTOMER_ID NUMBER(6) NOT NULL,
   FOREIGN KEY (INDIVIDUAL_CUSTOMER_ID) REFERENCES INDIVIDUAL_CLIENT(CUSTOMER_ID)
);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	BILL_ID	NUMBER(10,0)	No	(null)	1	(null)
2	BILL_AMOUNT	NUMBER(6,0)	No	(null)	2	(null)
3	INDIVIDUAL_CUSTOMER_ID	NUMBER(6,0)	No	(null)	3	(null)

17.

```
CREATE TABLE iNDIVIDUAL_PAYMENT(
   PAYMENT_ID NUMBER(10) NOT NULL PRIMARY KEY,
   PAYMENT_AMOUNT NUMBER(6) NOT NULL CHECK(PAYMENT_AMOUNT >= 0),
   PAYMENT_DATE DATE NOT NULL,
   PAYMENT_TIME TIMESTAMP NOT NULL,
   INDIVIDUAL_CUSTOMER_ID NUMBER(6) NOT NULL,
   BILL_ID NUMBER(10) NOT NULL,
   FOREIGN KEY (INDIVIDUAL_CUSTOMER_ID) REFERENCES INDIVIDUAL_CLIENT(CUSTOMER_ID),
   FOREIGN KEY (BILL_ID) REFERENCES INDIVIDUAL_CUST_BILL_DETAILS(BILL_ID)
);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	PAYMENT_ID	NUMBER(10,0)	No	(null)	1	(null)
2	PAYMENT_AMOUNT	NUMBER(6,0)	No	(null)	2	(null)
3	PAYMENT_DATE	DATE	No	(null)	3	(null)
4	PAYMENT_TIME	TIMESTAMP(6)	No	(null)	4	(null)
5	INDIVIDUAL_CUSTOMER_ID	NUMBER(6,0)	No	(null)	5	(null)
6	BILL_ID	NUMBER(10,0)	No	(null)	6	(null)

18.

```
CREATE TABLE INDIVIDUAL_CUST_REWARD(
   INDIVIDUAL_CUSTOMER_ID NUMBER(6) NOT NULL,
   MEMBER_NO NUMBER(10) NOT NULL,
   PRIMARY KEY (INDIVIDUAL_CUSTOMER_ID, MEMBER_NO),
   FOREIGN KEY (INDIVIDUAL_CUSTOMER_ID) REFERENCES INDIVIDUAL_CLIENT(CUSTOMER_ID)
);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	INDIVIDUAL_CUSTOMER_ID	NUMBER(6,0)	No	(null)	1	(null)
2	MEMBER_NO	NUMBER(10,0)	No	(null)	2	(null)

19.

```
CREATE TABLE HOTEL_ROOMS(
  ROOM_NO NUMBER(4) NOT NULL PRIMARY KEY,
  BED_TYPE VARCHAR(10) NOT NULL,
  ROOM_TYPE VARCHAR(10) NOT NULL,
  PRICE NUMBER(10) NOT NULL CHECK(PRICE >= 0),
  ROOM_AVAILABILITY VARCHAR(5) CHECK(ROOM_AVAILABILITY IN ('yes','no')),
  HOUSEKEEPING_ID NUMBER(9) NOT NULL,
  FOREIGN KEY (HOUSEKEEPING_ID) REFERENCES HOUSEKEEPING_CONCIERGE(EMPLOYEE_ID)
);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ROOM_NO	NUMBER(4,0)	No	(null)	1	(null)
2	BED_TYPE	VARCHAR2(10 BYTE)	No	(null)	2	(null)
3	ROOM_TYPE	VARCHAR2(10 BYTE)	No	(null)	3	(null)
4	PRICE	NUMBER(10,0)	No	(null)	4	(null)
5	ROOM_AVAILABILITY	VARCHAR2(5 BYTE)	Yes	(null)	5	(null)
6	HOUSEKEEPING_ID	NUMBER(9,0)	No	(null)	6	(null)

20.

```
CREATE TABLE BOOKING_DETAILS (
```

```
   INDIVIDUAL_CUST_ID NUMBER(6) NOT NULL,
   ROOM_NO NUMBER(4) NOT NULL,
   FOREIGN KEY (INDIVIDUAL_CUST_ID) REFERENCES INDIVIDUAL_CLIENT(CUSTOMER_ID),
   FOREIGN KEY (ROOM_NO) REFERENCES HOTEL_ROOMS(ROOM_NO),
   PRIMARY KEY (INDIVIDUAL_CUST_ID,ROOM_NO)
);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	INDIVIDUAL_CUST_ID	NUMBER(6,0)	No	(null)	1	(null)
2	ROOM_NO	NUMBER(4,0)	No	(null)	2	(null)

21.

```
CREATE TABLE EVENT_HOST (
   ORGANIZATIONAL_CLIENT_ID NUMBER(6) NOT NULL,
   EVENT_ID NUMBER(4) NOT NULL,
   FOREIGN KEY (ORGANIZATIONAL_CLIENT_ID) REFERENCES ORGANIZATIONAL_CLIENT(CUSTOMER_ID),
   FOREIGN KEY (EVENT_ID) REFERENCES EVENT(EVENT_ID),
   PRIMARY KEY (ORGANIZATIONAL_CLIENT_ID,EVENT_ID)
);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ORGANIZATIONAL_CLIENT_ID	NUMBER(6,0)	No	(null)	1	(null)
2	EVENT_ID	NUMBER(4,0)	No	(null)	2	(null)

22.

```
CREATE TABLE DIRECT_BILL_ACCOUNT (
   ORGANIZATIONAL_CLIENT_ID NUMBER(6) NOT NULL,
   ACCOUNT_NO NUMBER(10) NOT NULL,
   FOREIGN KEY (ORGANIZATIONAL_CLIENT_ID) REFERENCES ORGANIZATIONAL_CLIENT(CUSTOMER_ID),
   PRIMARY KEY (ORGANIZATIONAL_CLIENT_ID,ACCOUNT_NO)
);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ORGANIZATIONAL_CLIENT_ID	NUMBER(6,0)	No	(null)	1	(null)
2	ACCOUNT_NO	NUMBER(10,0)	No	(null)	2	(null)

23.

```
CREATE TABLE ORGANIZATION_CLIENT_BILL(
   BILL_ID NUMBER(10) NOT NULL PRIMARY KEY,
   AMOUNT NUMBER(10) CHECK(AMOUNT>=0),
   BILL_DATE DATE NOT NULL,
   ACCOUNTANT_ID NUMBER(9) NOT NULL,
   EVENT_ID NUMBER(4) NOT NULL,
```

```
  FOREIGN KEY (ACCOUNTANT_ID) REFERENCES ACCOUNTANT(EMPLOYEE_ID),
  FOREIGN KEY (EVENT_ID) REFERENCES EVENT(EVENT_ID)
);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	BILL_ID	NUMBER(10,0)	No	(null)	1	(null)
2	AMOUNT	NUMBER(10,0)	Yes	(null)	2	(null)
3	BILL_DATE	DATE	No	(null)	3	(null)
4	ACCOUNTANT_ID	NUMBER(9,0)	No	(null)	4	(null)
5	EVENT_ID	NUMBER(4,0)	No	(null)	5	(null)

24.

```
CREATE TABLE ORGANIZATIONAL_PAYMENT(
  PAYMENT_ID NUMBER(10) NOT NULL PRIMARY KEY,
  PAYMENT_MODE VARCHAR(20) NOT NULL,
  PAYMENT_AMOUNT NUMBER(6) NOT NULL CHECK(PAYMENT_AMOUNT >= 0),
  PAYMENT_DATE DATE NOT NULL,
  PAYMENT_TIME TIMESTAMP NOT NULL,
  ORG_CUSTOMER_ID NUMBER(6) NOT NULL,
  FOREIGN KEY (ORG_CUSTOMER_ID) REFERENCES ORGANIZATIONAL_CLIENT(CUSTOMER_ID)
);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	PAYMENT_ID	NUMBER(10,0)	No	(null)	1	(null)
2	PAYMENT_MODE	VARCHAR2(20 BYTE)	No	(null)	2	(null)
3	PAYMENT_AMOUNT	NUMBER(6,0)	No	(null)	3	(null)
4	PAYMENT_DATE	DATE	No	(null)	4	(null)
5	PAYMENT_TIME	TIMESTAMP(6)	No	(null)	5	(null)
6	ORG_CUSTOMER_ID	NUMBER(6,0)	No	(null)	6	(null)

25.

```
CREATE TABLE PAYMENT_THROUGH_DETAILS (
  PAYMENT_ID NUMBER(10) NOT NULL,
  BILL_ID NUMBER(10) NOT NULL,
  FOREIGN KEY (PAYMENT_ID) REFERENCES ORGANIZATIONAL_PAYMENT(PAYMENT_ID),
  FOREIGN KEY (BILL_ID) REFERENCES ORGANIZATION_CLIENT_BILL(BILL_ID),
  PRIMARY KEY (PAYMENT_ID,BILL_ID)
);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	PAYMENT_ID	NUMBER(10,0)	No	(null)	1	(null)
2	BILL_ID	NUMBER(10,0)	No	(null)	2	(null)

26.

```
CREATE TABLE INDIVIDUAL_PHONE (
   INDIVIDUAL_CUST_ID NUMBER(6) NOT NULL,
   PHONE_NO NUMBER(10) NOT NULL,
   FOREIGN KEY (INDIVIDUAL_CUST_ID) REFERENCES INDIVIDUAL_CLIENT(CUSTOMER_ID),
   PRIMARY KEY (INDIVIDUAL_CUST_ID,PHONE_NO)
);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	INDIVIDUAL_CUST_ID	NUMBER(6,0)	No	(null)	1	(null)
2	PHONE_NO	NUMBER(10,0)	No	(null)	2	(null)

4. Conclusion

In this report we discussed and drew the relational schemas for Database of Hotel Management System. We also give the data type and format for each attribute in each schema. Then we explain our assumptions in the documentation. This report analyzed the logical model of Database. The next step is to implement this database. In the future, we may change some design when facing practical difficulties and other requests.