Documentation

Assignment 1

Introduction/ Application Features/ Project Description

This hotel management database system will manage rooms and room reservations in our hotel. Guests will be able to view information about their booking, including room details and packages; hotel employees will be able to look up reservations linked with the respective guest.

System Description

The hotel management DBMS is an integrated system that links different components of hotels together. This system allows management of guest reservations, guests, and rooms. Our database is normalized and features multiple foreign key relationships to prevent data repetition and inconsistencies.

Our system first defines hotels each with a unique ID and its other relevant information such as hotel name and address. We then define the Room entity where we save relevant information about rooms including the hotel in which the rooms exist, linking it with the hotel ID as a foreign key. Next we define our Guest entity, defining unique guests and storing relevant information such as name and email address. With these entities defined, we can define our Reservation entity, which is used to reserve a room for a specific period of time for a specific guest. In order to meet these requirements, we store the Room_ID of the room being reserved as well as the Guest_ID of the guest who is reserving the room. Therefore, directly from the reservation entity we can find information about the room, guest, and even hotel by simply following foreign key relationships. We further go on to define a payment entity containing a foreign key to a reservation_ID. This relationship is useful as it provides flexibility in payment for reservations. Lastly, we defined the employee entity, where employee information can be stored, as can their specific role in the hotel as a foreign key relationship.

With this database architecture we are able to leverage control; for example in the form of only allowing unique Room_IDs to be saved as a foreign key in active reservations as well as flexibility for example in the form of multiple payments pointing to the same reservation. We also have no data repetition, and so we can be confident of the integrity of our data.

Entity Table

HOTEL

Hotel_ID*	Hotel_Name	Street_Name	City	Province	Postal_Code	Country	Rating
4ec3dfac	Four Seasons Hotel	60 Yorkville Ave	Toronto	Ontario	M4W 0A4	Canada	4.7
7550f5b1	Bisha Hotel	80 Blue Jays Way	Toronto	Ontario	M5V 2G3	Canada	4.5

ROOM

Room_ID*	**Hotel_ID**	Room_Number	Room_Price	Number_Of_Beds	Bed_Size	Breakfast_Included	Smoking	Accessibility
99bbdff3	4ec3dfac	*213*	640	1	King	TRUE	TRUE	TRUE
3db79a80	7550f5b1	*425*	515	1	King	TRUE	TRUE	TRUE

GUEST

Guest_ID*	First_Name	Last_Name	Email	Phone_Number
1ba53597	Nicholas	French	nich.f@gmail.com	819 486 1054
088718ac	Sydney	Walker	Syd.w@gmail.com	204 752 6574

RESERVATION

Reservation_ID*	**Room_ID**	**Guest_ID**	Check_in_Date	Check_out_Date	Number_of_Guests	Status
d0e6bec7	99bbdff3	1ba53597	09/03/2022	09/05/2022	2	BOOKED
8a1816a9	3db79a80	088718ac	11/06/2022	11/08/2022	2	PENDING

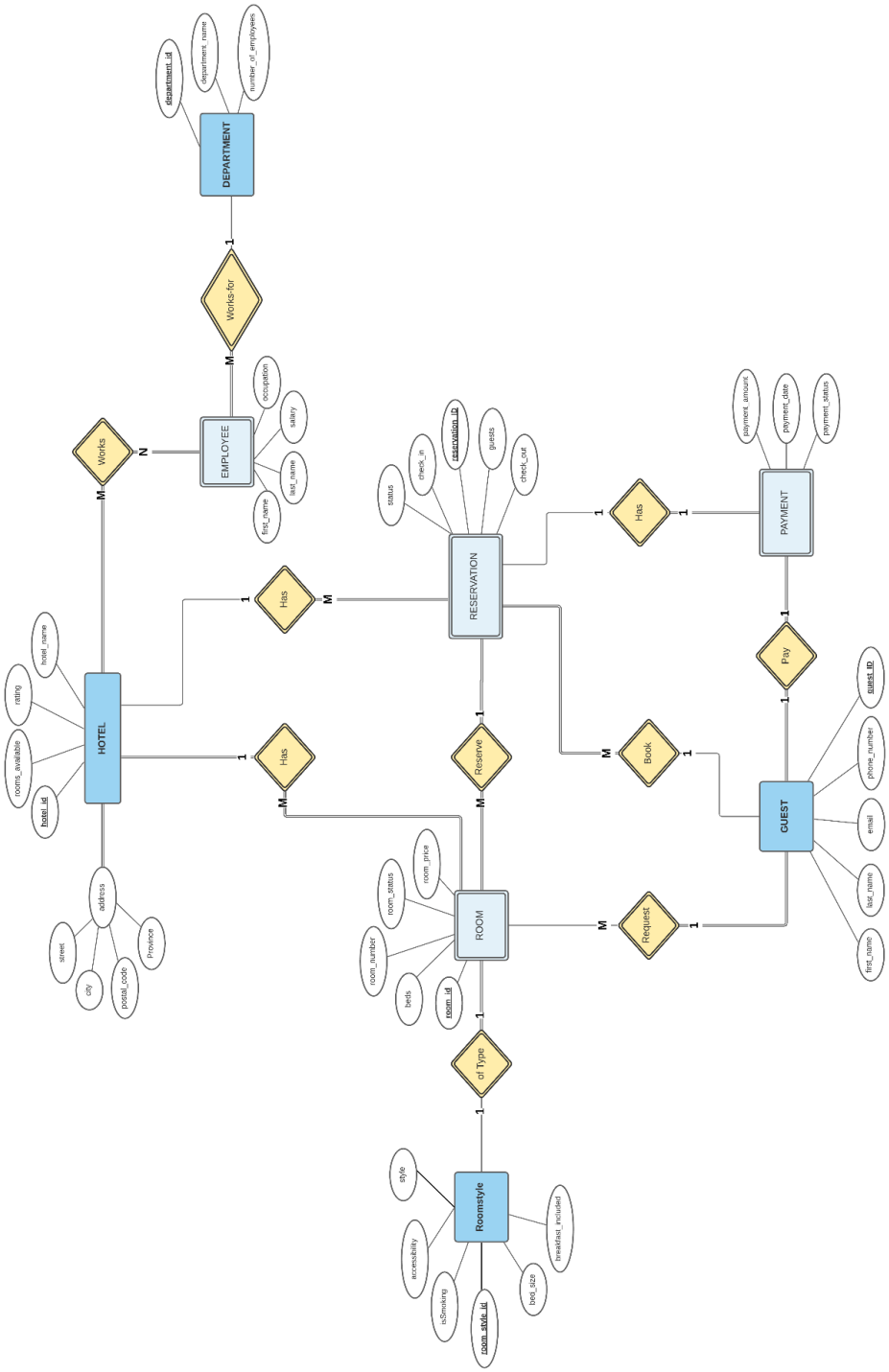
PAYMENT

Payment_ID*	**Reservation_ID**	Payment_Amount	Payment_Date
9aa36742	d0e6bec7	640	09/01/2022
a1badb00	8a1816a9	515	11/02/2022

EMPLOYEE

Employee_ID*	First_Name	Last_Name	Salary	**Hotel_ID**
315201f9	Tracey	Long	48290	4ec3dfac
6aee7bce	Jerry	Morin	45830	7550f5b1

Assignment 2



Assignment 3, 4 & 5 - SQL Files

```
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> SQL> SQL> SQL> SQL> SQL> SQL>   2    3
"FIRST_NAME","LAST_NAME","GUEST_ID","GUESTS"
"Alick","Lazare",979625,4
"Jessica","Singh",983492,4
"Anant","Jawanda",979632,4
"Gary","Holand",236222,4
"Alick","Lazare",979625,4
"Jessica","Singh",983492,4
"Anant","Jawanda",979632,4
"Gary","Holand",236222,4

8 rows selected.

SQL> SQL> SQL> SQL>   2    3    4
"HOTEL_ID","ROOMS_AVAILABLE","RATING","HOTEL_NAME","STREET_NAME","CITY","POSTAL_CODE","PROVINCE"
584983,97,4.3,"Test Hotel","Bloor and Yonge","Toronto","5JX65E","ON"
523657,70,4.2,"Hotel B","Jarvis Street","Toronto","G5TL1S","ON"

SQL> SQL> SQL>   2    3
"ROOM_STYLE_ID","ROOM_STYLE","BED_SIZE","INCLUDESBREAKFAST","ISSMOKING","ISACCESSIBILE"
688088,"Deluxe","King",1,0,1
789203,"Deluxe","King",1,0,1
345782,"Suite","King",1,0,0
903485,"Deluxe","King",1,1,1
698745,"Suite","King",1,1,1

SQL> SQL> SQL>   2    3    4
"DEPARTMENT_ID","HOTEL_ID","DEPARTMENT_NAME","NUMBER_OF_EMPLOYEES"
902746,584983,"Accounts",3
583999,523657,"Marketing",4
902744,584983,"Administration",5
902745,584983,"Customer Service",5

SQL> SQL> SQL>   2    3    4
"FIRST_NAME","LAST_NAME"
"Jempo","Hempo"
"Jerry","Fisher"

SQL> SQL> SQL>   2    3    4
"HOTEL_ID","ROOM_NUMBER"
523657,24
584983,56
584983,90

SQL> SQL> SQL>   2    3    4
"RESERVATION_ID","ROOM_ID","GUEST_ID","STATUS","CHECK_IN","CHECK_OUT","GUESTS"
513419,390533,979632,"Completed","16-NOV-21","20-NOV-21",4
632112,889984,236222,"Completed","01-AUG-22","03-AUG-22",4

SQL> SQL> SQL>   2    3
"GUEST_ID","RESERVATION_ID","PAYMENT_DATE","PAYMENT_AMOUNT","PAYMENT_STATUS"
979625,513418,"01-DEC-21",320.25,"Pending"

SQL> SQL>   2    3    4
"PAYMENT_DATE","TOTALPAYMENTS"
"01-DEC-21",320.25
"20-JUL-22",265.24
"10-NOV-21",346.25

SQL> SQL> SQL>   2    3    4
"HOTEL_NAME","DEPARTMENT_NAME"
"Test Hotel","Maintainance"
"Test Hotel","Administration"
"Test Hotel","Customer Service"
"Test Hotel","Accounts"
"Hotel B","Culinary"
"Hotel B","Marketing"
```

```
SQL> SQL>   2    3    4    5
"DEPARTMENT_NAME","FIRST_NAME","LAST_NAME","OCCUPATION","HOURLY_WAGE"
"Maintainance","Alick","Lazare","Janitor",25.6
"Maintainance","Emily","Jenn","Janitor",25.6
"Administration","John","Doe","Manager",36.25
"Customer Service","Anna","Sara","Receptionist",20.8
"Culinary","Jempo","Hempo","Chef",43.1
"Culinary","Jerry","Fisher","Head Chef",25.6
"Marketing","John","James","Math guy",25.6
"Marketing","Gary","Wary","Consultant",25.6

8 rows selected.

SQL> SQL>   2    3    4
"GUEST_ID","ROOM_ID","PAYMENT_AMOUNT"
979632,390533,346.25
979625,327583,320.25
236222,889984,265.24

SQL> SQL>   2    3
View created.

SQL>
"FIRST_NAME","LAST_NAME","OCCUPATION","EMPLOYEE_ID"
"Alick","Lazare","Janitor",638156
"Emily","Jenn","Janitor",834953
"John","Doe","Manager",903452
"Anna","Sara","Receptionist",903622
"Bob","Joe","Greeter",123953
"Jempo","Hempo","Chef",444555
"Jerry","Fisher","Head Chef",211659
"John","James","Math guy",296325
"Gary","Wary","Consultant",122612

9 rows selected.

SQL> SQL>   2    3    4
View created.

SQL>
"HOTEL_ID","ROOM_NUMBER","BEDS","ROOM_STATUS"
584983,56,1,"needs cleaning"
523657,24,2,"needs cleaning"

SQL> SQL>   2    3    4    5    6    7
View created.

SQL>
"FIRST_NAME","GUEST_ID","LAST_NAME"
"Anant",979632,"Jawanda"

SQL> SQL> SQL> SQL>   2    3    4    5    6
"GUEST_ID","'RESERVED'"
236222,"payment-made"
236222,"reserved"
979625,"payment-made"
979625,"reserved"
979632,"reserved"

SQL> SQL> SQL>   2    3    4    5    6    7
"GUEST_ID","FIRST_NAME","LAST_NAME","EMAIL","PHONE_NUMBER"
979632,"Anant","Jawanda","anant@ryerson.ca","6478934567"
979625,"Alick","Lazare","alazare@ryerson.ca","6471234567"
236222,"Gary","Holand","gholand@gmail.ca","6473457623"

SQL> SQL> SQL>   2    3    4    5    6
"RESERVATION_ID","GUEST_ID","ROOM_ID","CHECK_IN"
513418,979625,327583,"31-DEC-21"

SQL> SQL> SQL> SQL>   2    3    4
"CHECK_IN","NUMBER_OF_RSVP"
"31-DEC-21",2

SQL> SQL> SQL>   2    3    4    5
"TOTAL_REVENUE","PAYMENT_DATE"
346.25,"10-NOV-21"

SQL> SQL> SQL> Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
```

Assignment 6

Functional Dependencies for Hotel Management

TABLE HOTEL

Hotel Table with PK: hotel_ID

Dependencies

hotel_ID → hotel_name
hotel_ID → street_name
hotel_ID → city
hotel_ID → province
hotel_ID → postal_code
hotel_ID → rating
hotel_ID → rooms_available

Hotel_ID*	Hotel_Name	Street_Name	City	Province	Postal_Code	Rating	Rooms_Avilable

TABLE ROOM

Room Table with PK: room_ID
Foreign key FK: hotel_id
Foreign key FK: roomstyle_id

Dependencies

Room_id → hotel_id
Room_id → roomstyle_id
Room_id → beds
Room_id → room_number
Room_id → room_status
Room_id → room_price

Room_ID*	Hotel_ID	Room_style_ID	Beds	Room_number	Room_status	Room_price

TABLE ROOMSTYLE

Roomstyle Table with PK: roomstyle_id

Dependencies

Room_Style_ID → style
Room_Style_ID → bed_size
Room_Style_ID → breakfast_included
Room_Style_ID → isSmoking
Room_Style_ID → accessibility

Room_Style_ID*	Style	Bed_Size	Breakfast_Included	isSmoking	Accessibility

TABLE GUEST

Guest Table with PK: guest_id

Dependencies

Guest_id → first_name
Guest_id → last_name
Guest_id → email
Guest_id → phone_number

Guest_ID*	First_Name	Last_Name	Email	Phone_Number

TABLE RESERVATION

Reservation Table with PK: reservation_id
Foreign key FK: room_id
Foreign key FK: guest_id

Dependencies

Reservation_id → room_id
Reservation_id → guest_id
Reservation_id → check_in
Reservation_id → check_out
Reservation_id → number_of_guests

Reservation_id → status

Reservation_ID*	Room_ID	Guest_ID	Check_in	Check_out	Number_of_Guests	Status

TABLE PAYMENT

Payment Table with Foreign Key FK: guest_id,reservation_id

Dependencies

{guest_id, reservation_id} → payment_amount
{guest_id, reservation_id} → payment_date
{guest_id, reservation_id} → payment_status

Guest_id	Reservation_ID	Payment_Amount	Payment_Date	Pyament_Status

TABLE EMPLOYEE

Employee Table with PK: employee_id
Foreign key FK: Department_id

Dependencies

Employee_id → department_id
Employee_id → first_name
Employee_id → last_name
Employee_id → hourly_wage
Employee_id → occupation

Employee_ID*	Department_ID	First_Name	Last_Name	hourly_wage	occupation

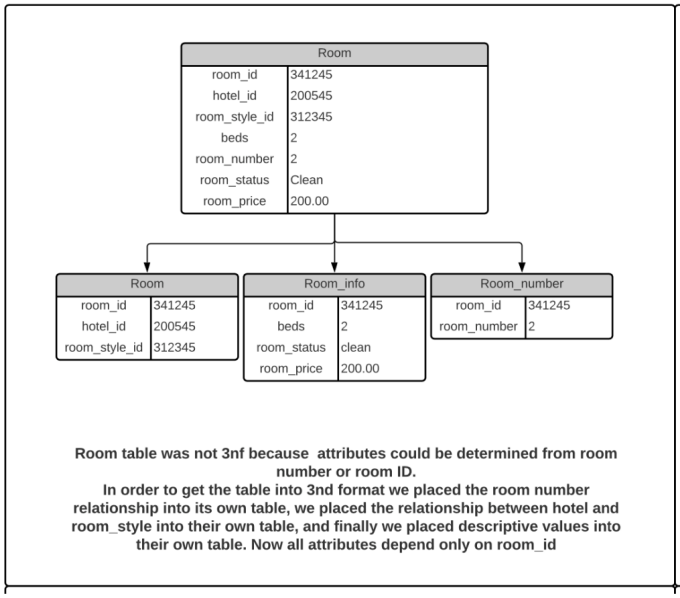
TABLE DEPARTMENT

Department Table with PK: department_id
Foreign key FK: Hotel_id

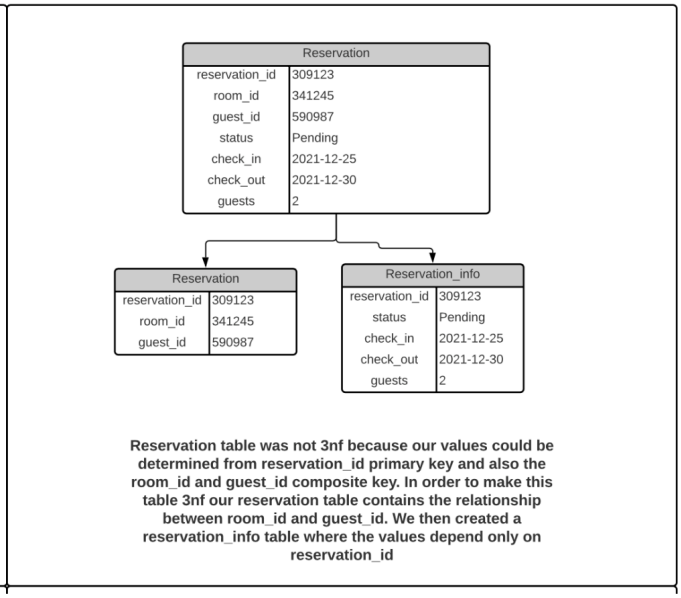
Dependencies
Department_id → hotel_id
Department_id → department_name
Department_id → number_employees

Department_ID*	Hotel_ID	Department_name	Number_of_Employees

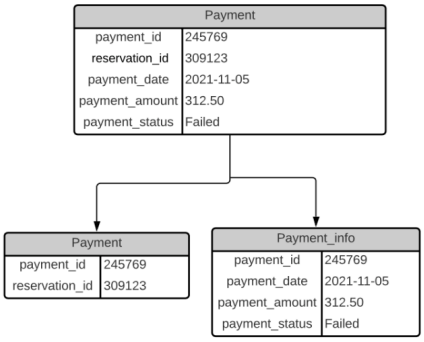
Assignment 7



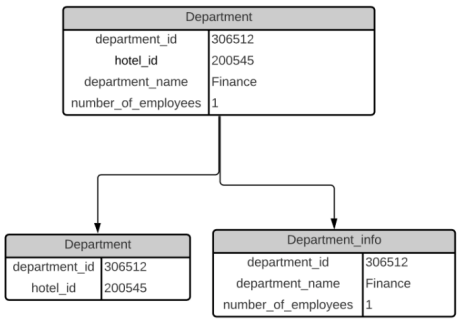
Room table was not 3nf because attributes could be determined from room number or room ID.
In order to get the table into 3nd format we placed the room number relationship into its own table, we placed the relationship between hotel and room_style into their own table, and finally we placed descriptive values into their own table. Now all attributes depend only on room_id



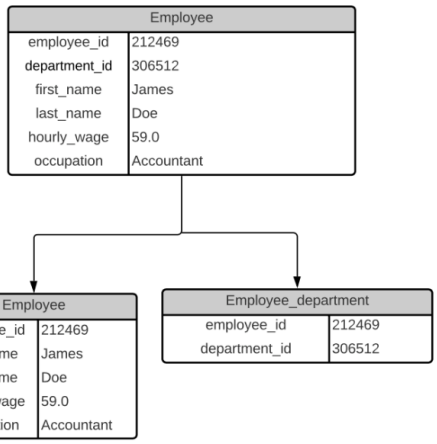
Reservation table was not 3nf because our values could be determined from reservation_id primary key and also the room_id and guest_id composite key. In order to make this table 3nf our reservation table contains the relationship between room_id and guest_id. We then created a reservation_info table where the values depend only on reservation_id



The payment table previously had a composite primary key the guestID and reservationID, this made all other fields partially dependent on these two fields, meaning that the table was not in 2nf form. In order to get the table to 2nf and 3nf form we created the payment_id field which acts as the primary key for the payment table. The payment table then describes the relationship between payment and reservation through the reservation_id. We then created the payment_info table with primary key as the payment_id. All fields in the payment_info are dependent on only the payment_id, making these tables 3nf.



The department table previously contained all fields related to the department. Some of these fields could also be dependent on hotel_id. We split the department table into two, with the department table capturing the relationship between the department and hotel through hotel_id foreign key. They department_info describes the department with primary key department_id again, making all fields functionally dependent on only the department_id



The employee table was not 3nf because some fields could be determined from the department_id and occupation fields. In order to get this table 3nf compliant we split it into the employee_department table which contains the relationship between employee_id(pk) and department_id(fk). We then define details about employe in the employee table where all fields are functionally dependent on employee_id(pk) making these tables 3nf.

Assignment 8

BCNF

Reservation	
reservation_id	309123
room_id	341245
guest_id	590987

Reservation_info	
reservation_id	309123
status	Pending
check_in	2021-12-25
check_out	2021-12-30
guests	2

FDs TABLE reservation
reservation_id--> room_id
reservation_id--> guest_id

FDs TABLE reservation_info
reservation_id --> status
reservation_id --> check_in
reservation_id --> check_out
reservation_id --> guests

Bernstein's Algorithm:
1) List of attributes and FDs are shown above
2) Tables were decomposed to have minimal list of FDs
3) Candidate keys
reservation_info: "reservation_id"
4) Room_info
R1(reservation_id, status)
R2(reservation_id, check_in)
R3(reservation_id, check_out)
R4(reservation_id, guests)

BCNF

Payment	
payment_id	245769
reservation_id	309123

Payment_info	
payment_id	245769
payment_date	2021-11-05
payment_amount	312.50
payment_status	Failed

FDs TABLE payment_info
payment_id--> payment_date
payment_id--> payment_amount
payment_id --> payment_status

Bernstein's Algorithm:
1) List of attributes and FDs are shown above
2) Tables were decomposed to have minimal list of FDs
3) Candidate keys
payment_info: "payment_id"
4) Room_info
R1(payment_id, payment_date)
R2(payment_id, payment_amount)
R3(payment_id, payment_status)

3NF

Department	
department_id	306512
hotel_id	200545

Department_info	
department_id	306512
department_name	Finance
number_of_employees	1

BCNF

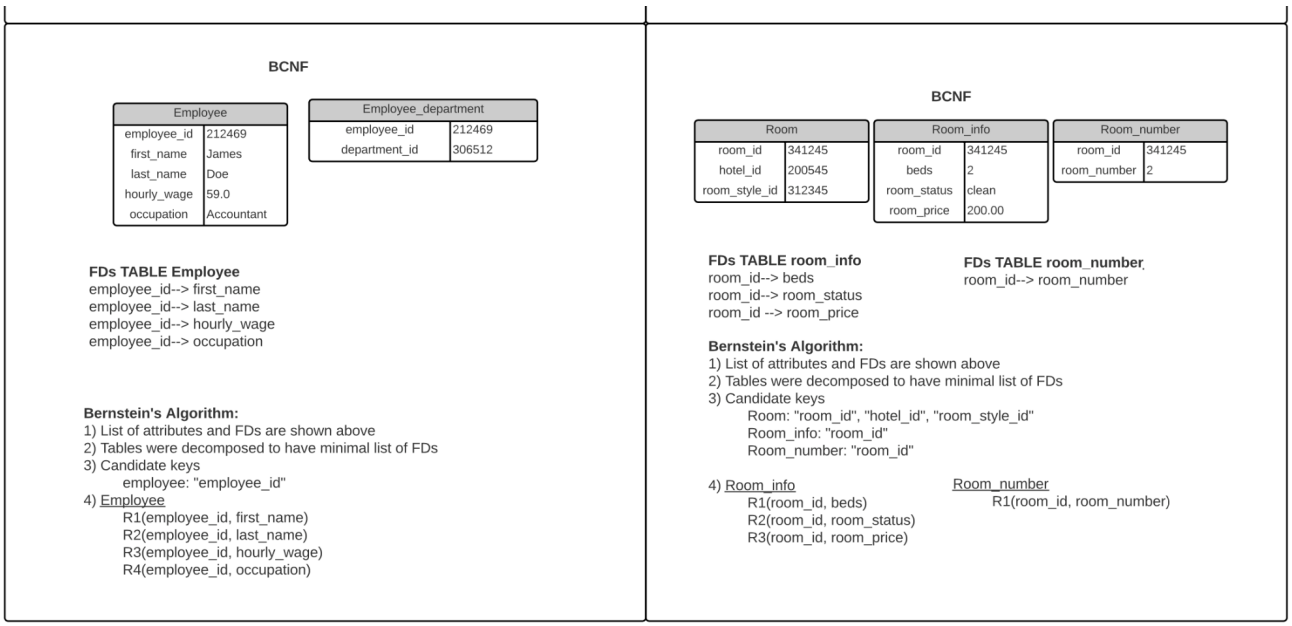
Hotel Department	
department_id	306512
hotel_id	200545

Department	
department_id	306512
d_name_id	1234

Department Name	
d_name_id	1234
department_name	Finance
number_of_employees	1

FDs TABLE Department_name
d_name_id--> department_name
d_name_id--> number_of_employees

Bernstein's Algorithm:
1) List of attributes and FDs are shown above
2) Tables were decomposed to have minimal list of FDs
3) Candidate keys
department_name: "d_name_id"
4) Department_name
R1(d_name_id, department_name)
R2(d_name_id, number_of_employees)



Assignment 10

List of Queries with RA notation

SELECT * Hotel WHERE rating >= 4 ORDER BY rating DESC; ↓ $\sigma_{\text{rating} >= 4}$ (HOTEL)
SELECT * from payment_info WHERE payment_amount >= 300 AND payment_status = ‘Pending’; ↓ $\sigma_{\text{payment\_amount} > 50000 \text{ AND payment\_status} = \text{“Pending”}}$ (PAYMENT_INFO)
SELECT * Reservation_info WHERE status = ‘pending’ AND guest<=2 ↓

$\sigma_{\text{payment\_status = “Pending” AND guest <= 2}}$ (RESERVATION_INFO)
SELECT * Department Name WHERE department_name = Finance OR d_name_id=1234 ↓ $\sigma_{\text{(department\_name = “Finance”) OR (d\_name\_id = 1234)}}$ (DEPARTMENT)
SELECT first_name, last_name FROM employee WHERE department_id=121111 ↓ $\pi_{\text{first\_name, last\_name}}\ \sigma_{\text{(department\_id = 121111)}}$ (EMPLOYEE)

Concluding Remarks on Design Experience

The design experience of building a Hotel Management Database Management System was a unique experience. We found that it was a great learning experience as we were able to apply our knowledge of the lecture content into actual applications in our labs. These lab sessions were incredibly useful especially with the immense walkthrough with the TA and the comprehensive feedback we received. In scenarios when our team was lost, the TA was able to help guide us through the next steps for our project. This allowed our team to collaboratively and strategically work through the pitfalls, and ultimately create a functional database from scratch.