Süleyman Berber 31293
CS306 Phase 2

```
CREATE TABLE Patients(
p_id INT,
gender varchar(1),
p_name varchar(50),
contact varchar(50),
bday DATE,
primary key (p_id)
);

Create table stays_in_room(
p_id INT,
room_id INT,
stay_in DATE,
primary key (p_id, room_id),
Foreign Key (p_id) References Patients (p_id) ON DELETE CASCADE,
Foreign Key (room_id) References Rooms (room_id) ON DELETE CASCADE
);
```

Patients Table: p_id → gender, p_name, contact, bday

Explanation: The patient ID (p_id) uniquely determines the patient's gender, name, contact information, and birth date. Each patient ID corresponds to exactly one set of gender, name, contact, and birth date. .

stays_in_room Table: (p_id, room_id) → stay_in

Explanation: Each combination of patient ID (p_id) and room ID (room_id) uniquely determines the stay-in date (stay_in). This means that for each patient and room, there is a specific date on which they stayed in that room.

Patient Table:
Functional Dependencies:
p_id → gender, p_name, contact, bday gender,

In this table, the candidate key is {p_id}, which is also a key. Therefore, the table is in BCNF because every non-trivial functional dependency's left-hand side (LHS) is a key. The table is in BCNF. The only functional dependency is based on the primary key (p_id), and it determines all other attributes. There are no partial dependencies.

Conclusion: The Patients table is already in BCNF.

Stays_in_room Table:
Functional Dependencies:
(p_id, room_id) → stay_in

In this table, the (p_id, room_id), which is also a superkey. Therefore, the table is in BCNF because every non-trivial functional dependency's left-hand side (LHS) is a superkey. The table is in BCNF. (p_id, room_id) determines all other attributes.

Conclusion: The Stays_in_room table is already in BCNF.

BCNF Analysis

Patients Table: The table is in BCNF. The only functional dependency is based on the primary key (p_id), and it determines all other attributes. There are no partial dependencies.

stays_in_room Table: The table is in BCNF. (p_id, room_id) is super key and it determines all other attributes.

Since I have not decomposed both of the tables because they are in BCNF, I'll pick the original tables, Patient and stays_in_room for inserting rows.

1. Inserting into Patients table:

```
INSERT INTO Patients (p_id, gender, p_name, contact, bday) VALUES
(1, 'M', 'John Smith', 'john@gmail.com', '1990-05-15'),
(2, 'F', 'Emily Johnson', 'emily@gmail.com', '1988-09-22'),
(3, 'M', 'Michael Brown', 'michael@hotmail.com', '1992-03-10'),
(4, 'F', 'Jessica Williams', 'jessica@gmail.com', '1995-11-27'),
(5, 'F', 'William Taylor', 'william@hotmail.com', '1985-07-18'),
(6, 'F', 'Sophia Anderson', 'sophia@gmail.com', '1998-01-03'),
(7, 'M', 'Matthew Martinez', 'matthew@gmail.com', '1993-06-30'),
(8, 'F', 'Emma Garcia', 'emma@gmail.com', '1991-12-14'),
(9, 'M', 'Daniel Rodriguez', 'daniel@hotmail.com', '1987-04-28'),
(10, 'F', 'Olivia Hernandez', 'olivia@hotmail.com', '1994-08-09');
```

p_id	gender	p_name	contact	bday
1	M	John Smith	john@gmail.com	1990-05-15
2	F	Emily Johnson	emily@gmail.com	1988-09-22
3	M	Michael Brown	michael@hotmail.com	1992-03-10
4	F	Jessica Williams	jessica@gmail.com	1995-11-27
5	F	William Taylor	william@hotmail.com	1985-07-18
6	F	Sophia Anderson	sophia@gmail.com	1998-01-03
7	M	Matthew Martinez	matthew@gmail.com	1993-06-30
8	F	Emma Garcia	emma@gmail.com	1991-12-14
9	M	Daniel Rodriguez	daniel@hotmail.com	1987-04-28
10	F	Olivia Hernandez	olivia@hotmail.com	1994-08-09
NULL	NULL	NULL	NULL	NULL

2. Inserting into stays_in_room table:

INSERT INTO stays_in_room (p_id, room_id, stay_in) VALUES
(1, 101, '2024-04-23'),
(2, 102, '2024-04-22'),
(3, 103, '2024-04-21'),
(4, 104, '2024-04-20'),
(5, 102, '2024-04-19'),
(6, 106, '2024-04-18'),
(7, 107, '2024-04-17'),
(8, 104, '2024-04-16'),
(9, 109, '2024-04-15'),
(10, 110, '2024-04-14');

p_id	room_id	stay_in
1	101	2024-04-23
2	102	2024-04-22
3	103	2024-04-21
4	104	2024-04-20
5	102	2024-04-19
6	106	2024-04-18
7	107	2024-04-17
8	104	2024-04-16
9	109	2024-04-15
10	110	2024-04-14
NULL	NULL	NULL

First Query: Retrieve the names and contacts of patients who have a medical record in room 102.

Relational Algebra Equivalent:
πPatients.p_name,Patients.contact(σstays_in_room.room_id=102(Patients⋈stays_in_room))

```
81  SELECT Patients.p_name, Patients.contact
82  FROM Patients
83  JOIN stays_in_room ON Patients.p_id = stays_in_room.p_id
84  WHERE stays_in_room.room_id = 102;
```

100% 25:78

Result Grid Filter Rows: Search Export:

p_name	contact
Emily Johnson	emily@gmail.com
William Taylor	william@hotmail.com

Second Query: Calculate the total number of stays and the average age of patients for each room between April 15, 2024, and April 22, 2024.

```
113  SELECT stays_in_room.room_id,
114         COUNT(stays_in_room.p_id) AS total_stays,
115         AVG(YEAR(CURDATE()) - YEAR(Patients.bday)) AS average_age
116  FROM stays_in_room
117  JOIN Patients ON stays_in_room.p_id = Patients.p_id
118  WHERE stays_in_room.stay_in BETWEEN '2024-04-15' AND '2024-04-22'
119  GROUP BY stays_in_room.room_id;
```

100% 32:111

Result Grid Filter Rows: Search Export:

room_id	total_stays	average_age
102	2	37.5000
103	1	32.0000
104	2	31.0000
106	1	26.0000
107	1	31.0000
109	1	37.0000

I added a "check" constraint to the Patients table to ensure that the gender field only allows values 'M' for male and 'F' for female. The constraint will prevent any other values from being inserted into the gender field.

ALTER TABLE Patients
ADD CONSTRAINT chk_gender CHECK (gender = 'M' OR gender = 'F');

INSERT INTO Patients (p_id, gender, p_name, contact, bday) VALUES (11, 'X', 'Alex', 'alex@example.com', '1990-01-01');

When I try to insert the data, I get an error.
Error Code: 3819. Check constraint 'chk_gender' is violated.