Nazerke Kulan 19010322З

Task 1 (20 points)

Consider the following relation that stores information about students living in dormitories at a college:

`College (lastName, stuId, homeAdd, homePhone, dormRoom, roommateName, dormAdd, status, mealPlan, roomCharge, mealPlanCharge)`

Assume:

1) Each student is assigned to one dormitory room and may have several roommates.
2) Names of students are not unique.
3) The college has several dorms. `dormRoom` contains a code for the dorm and the number of the particular room assigned to the student. For example, A221 means Adams Hall, room 221. Dorm names are unique.
4) The `dormAdd` is the address of the dorm building. Each building has its own unique address. For example, Adams Hall may be 123 Main Street, Anytown, NY 10001.
5) `status` tells the student's status: `Freshman`, `Sophomore`, `Junior`, `Senior`, or `Graduate Student`.
6) `mealPlan` tells how many meals per week the student has chosen as part of his or her meal plan. Each meal plan has a single `mealPlanCharge` associated with it.
7) The `roomCharge` is different for different dorms, but all students in the same dorm pay the same amount.

Answer the following questions:

a. Using these assumptions and stating any others you need to make, list all the non-trivial functional dependencies for this relation.

My functional dependencies are:

{stuId} –> {lastName, homeAdd, homePhone, dormRoom, roommateName, dormAdd, status, mealPlan, roomCharge, mealPlanCharge}
mealPlan –> mealPlanCharge
dormRoom –> dormAdd
dormAdd –> roomCharge

b. What are the candidate keys for this relation? Identify the primary key.

The only candidate key that I could find here was stuId.
All other attributes composite or single can't determine any record uniquely.
Analogically our primary key here is stuId.

c. Is the relation in third normal form? If not, find a 3NF lossless join decomposition of College that preserves dependencies.

1NF: It is in first normal form, because every attribute is single-valued for each tuple.
2NF: Because relation is 1NF and the key consists of a single attribute(which is stuId) the relation is automatically 2NF.
3NF: The answer is not.

Definition: A relation is in **third normal form** (3NF) if, whenever a nontrivial functional dependency $X \rightarrow A$ exists, then either X is a superkey or A is a member of some candidate key.

Nazerke Kulan 190103223

Here as I showed, in task (a) we can see all of our nontrivial functional dependencies. None of A is a member of candidate key and none of X is a superkey. Which means that now it is not in 3NF.
`College (lastName, stuId, homeAdd, homePhone, dormRoom, roommateName, dormAdd, status, mealPlan, roomCharge, mealPlanCharge)`

NewCollege(stuId, lastName, status, dormRoom, homeAddress, homePhone, mealPlan)
Room(stuId, roommateName)
Meal(mealPlan, mealPlanCharge)
Dorm(dormRoom, dormAdd)
DormCharge(dormAdd, roomCharge)

stuId	lastName	status	dormRoom	homeAddress	homePhone	mealPlan
190103223	Nazerke	junior	A439	Imanova 70	7-25-79	Breakfast#5
190103584	Araylym	freshman	B238	Moldagulova 34	3-14-44	Dinner#21
190103574	Aizhan	sophomore	A418	Satpayev 21	7-09-99	Breakfast#6

stuId	roommateName
190103223	Dana
190103223	Dilnaz
190103574	Asel
190103584	Erkezhan

mealPlan	mealCharge
Breakfast#5	4$
Dinner#21	5$
Breakfast#6	3$

dormRoom	dormAddres
A439	Adam Hall
B238	Beakon Hall
A418	Adam Hall

dormAddress	roomCharge
Adam Hall	700$
Beakon Hall	550$

d. Is the relation or resulting set of relations in Boyce-Codd Normal Form? If not, find a lossless join decomposition that is in BCNF. Identify any functional dependencies that are not preserved.

3NF relations are BCNF if there is only one candidate key and the key is not composite.

So the answer is YES. As it satisfy the condition we can condisider the resulting set of relations as Boyce-Codd Normal Form.

Nazerke Kulan 19010322З

Task 2 (25 points)

Consider the following table that holds basic information about members of a club:

`ClubMember(memberId, lastName, firstName, telephone)`

a. Assume you need to store the date the member joined the club and the date the membership ended. Create and/or modify the table to store these dates.

```
    CREATE TABLE ClubMember(
      memberId int NOT NULL PRIMARY KEY,
      lastName varchar(255) NOT NULL,
      firstName varchar(255) NOT NULL,
      telephone char(11) UNIQUE,
      joined_date date NOT NULL,
      quitted_date date
);
```

b. Insert five records indicating that five members joined the club within the past year, all on different dates.

```
INSERT INTO ClubMember
VALUES
(190103195, 'Ibragim', 'Akmaral', 87051954953, '2020-07-14', '2020-08-19'),
(190103345, 'Mukashev', 'Sagdash', 87017653498, '2020-08-14', NULL),
(190103374, 'Otesh', 'Berikzhan', 87002394365, '2019-12-23', NULL),
(190103223, 'Kulan', 'Nazerke', 87002390794, '2019-11-23', NULL),
(190103352, 'Kulanova', 'Asylym', 87002390799, '2020-01-29', '2020-07-17');
```

c. Write and execute an SQL query to find all the members who belong to the club as of today.

```
SELECT *
FROM ClubMember
WHERE quitted_date IS NULL;
```

d. Modify the table to show that one member dropped his or her membership three months ago.

```
UPDATE ClubMember
SET quitted_date = DATEADD(month, -3, GETDATE())
WHERE memberId = 190103195;
```

e. Repeat the query in part (c) to demonstrate that the person who dropped membership is no longer an active member.

```
SELECT *
FROM ClubMember
WHERE quitted_date IS NOT NULL;
```