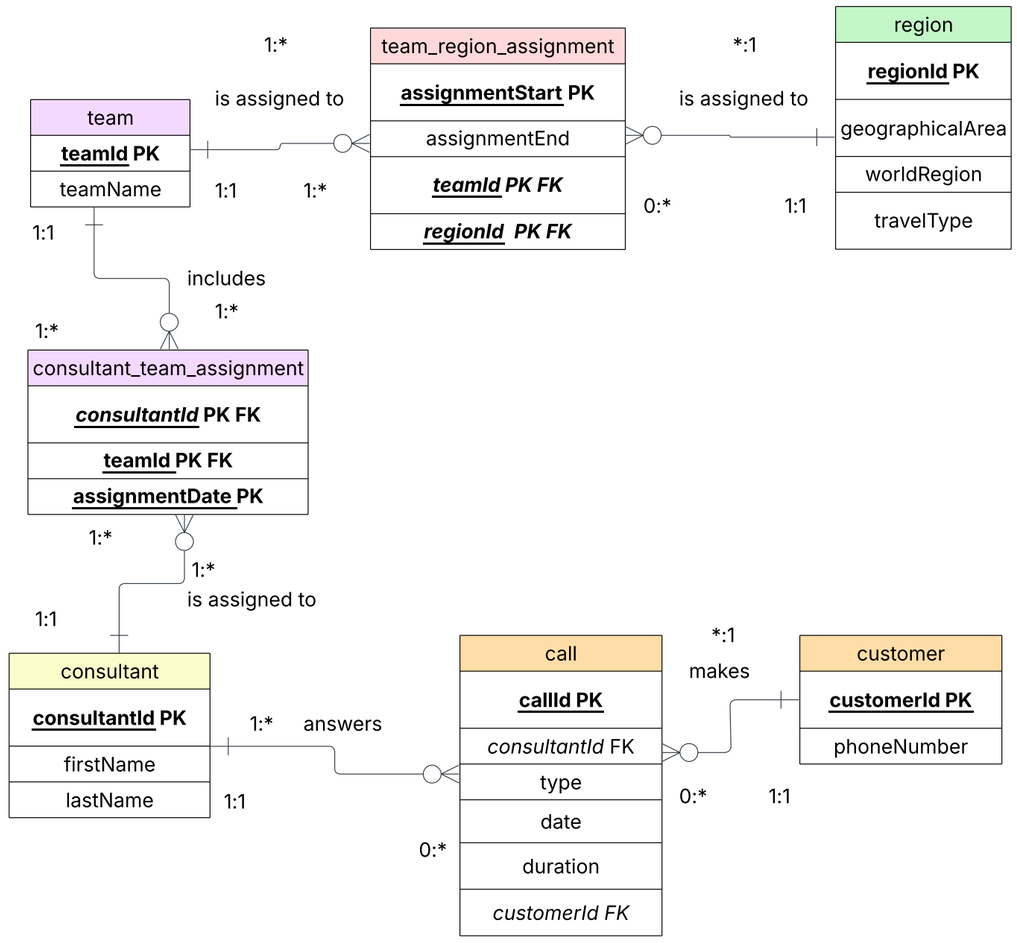
1)

travel consulting company ER Model

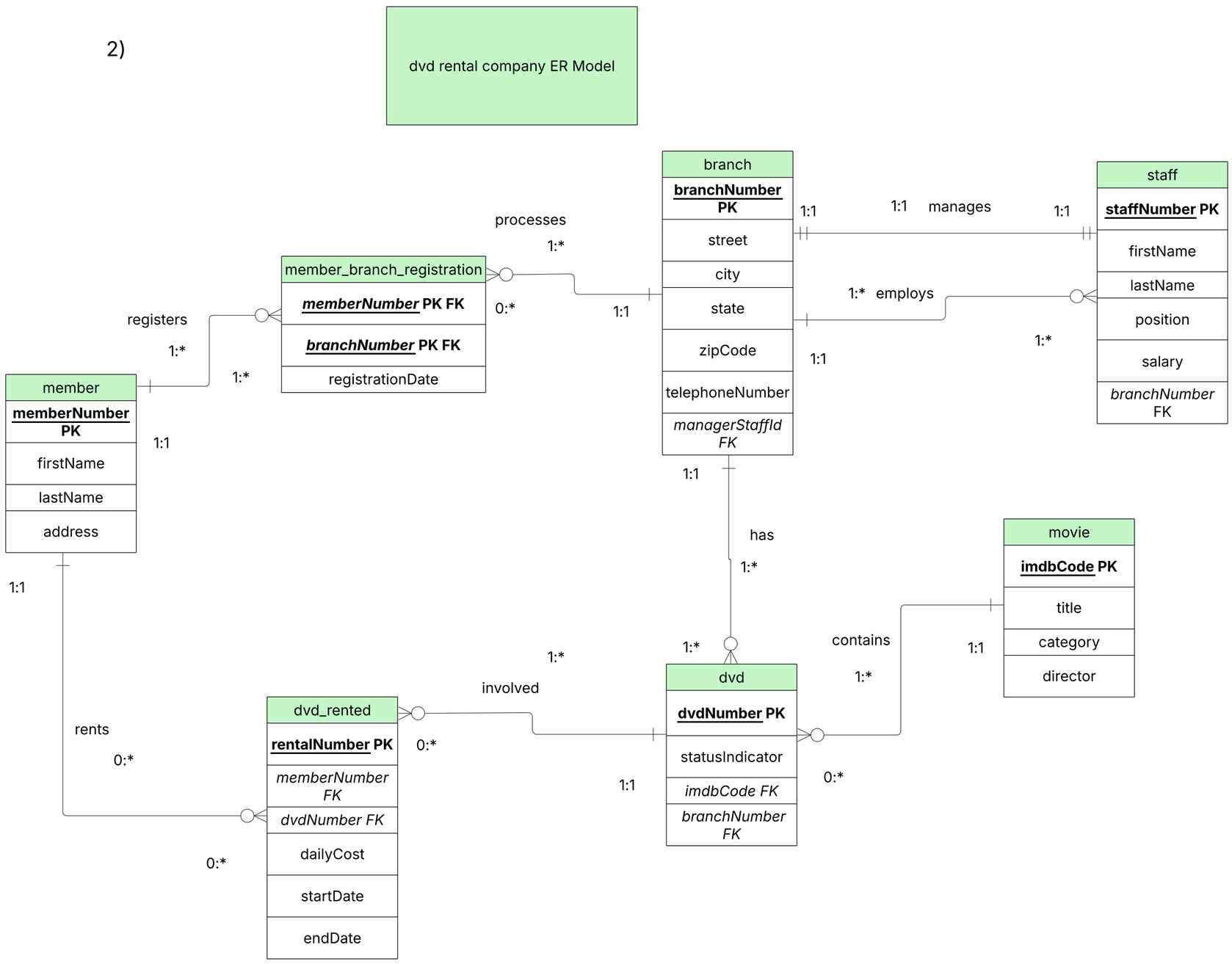


Assumptions:

1. A team member will only handle one region a time.
2. The team-region assignments as well as the consultant-team assignments are tracked over time.
3. Monthly statistics are calculated from call data.
4. Although specific customer attributes were not listed, I assumed a customer entity was needed to establish the relationship of the call log with calls that consultants receive from customers.
5. Specific attributes for consultants were not provided, so I included firstName and lastName to make each consultant identifiable and the table functional.
6. The guidelines state that management can reassign teams to different regions depending on business needs or the time of year, so I decided to create a tracking table called team_region_assignment. This has assignmentStart and assignmentEnd dates so teams can be assigned to various regions throughout the year but ensures that a team only is working with one region at a time.
7. The guidelines state that consultants can be assigned to other teams for coverage, so I created a joining table called consultant_team_assignment to track any overlapping team assignments. I included the assignmentDate so that if a consultant goes back and forth between teams, the database can still store each of these.

2)

dvd rental company ER Model



Assumptions:

1. The guidelines stated that the member's full name and the title of the DVD being rented should be held, so I chose to include the foreign keys of memberNumber and dvdNumber in the dvd_rented table to reference that information in order to avoid redundancy.
2. The guidelines said that staff name should be stored, so I split this into firstName and lastName to be consistent with how the member table stored names.
3. Members can register at multiple branches, so their registrationDate could not be stored in the member table. So, I created a bridging table called member_branch_registration to track specific dates that a member registered at a certain branch.
4. The constraint of limiting members to a maximum of 10 DVD rentals at a time was treated as a business level constraint, not a database constraint. I kept the cardinality between member and dvd_rented as 0:* to account for a history of any past rentals, so that a member would not be capped at 10 total rentals.
5. The staff table has a position attribute, which could be the role Manager, and I assumed that the one manager per branch constraint should be enforced at the database level. So, since a manager is a part of the staff, but each branch must have only one manager, I created a 1:1 relationship between branch and staff and added managerStaffId as a foreign key in the branch table. Thus, no two people at the same branch could both have positions set to Manager.
6. A staff member can only work at one branch at any given time.