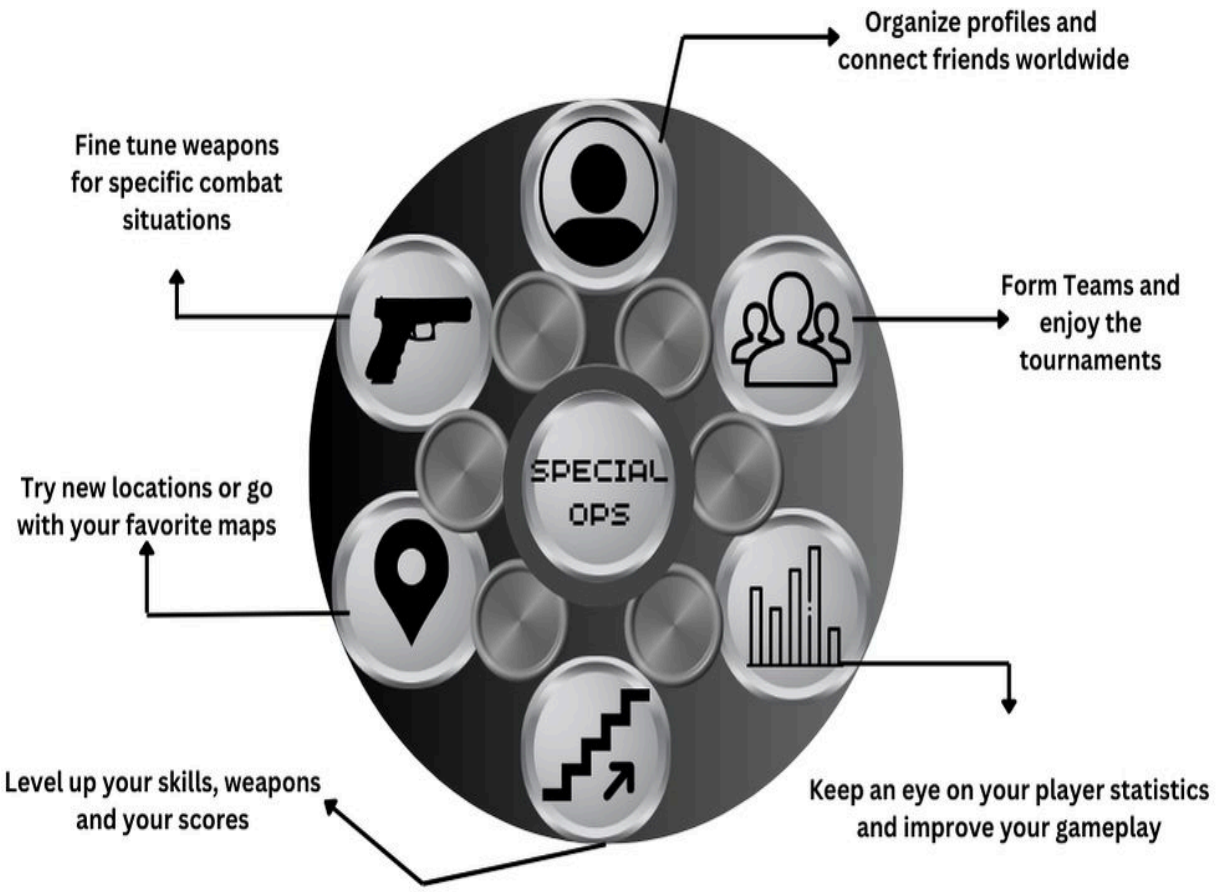
Special Ops

Team ID : T207

Objective

Call of Duty databases are well suited for India's thriving gaming sector. The number of Indians who like gaming and esports is increasing day by day, and with that, the demand for tools that improve gameplay, like comprehensive databases with user-generated material, techniques, and tips, is rising. This database has the potential to be extremely helpful in assisting the growing gaming community in India by offering insightful information and encouraging creativity and interaction.

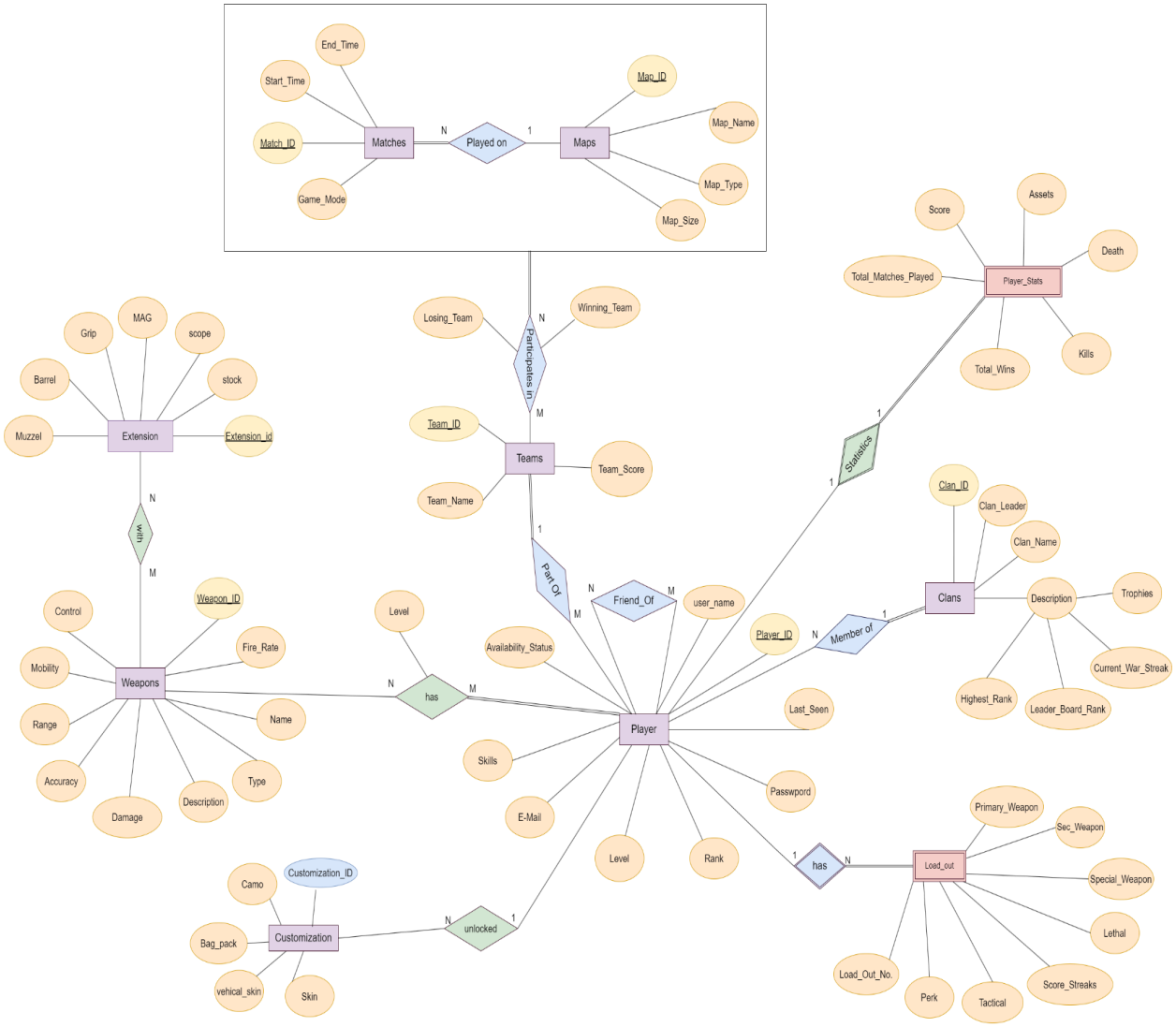


Goals

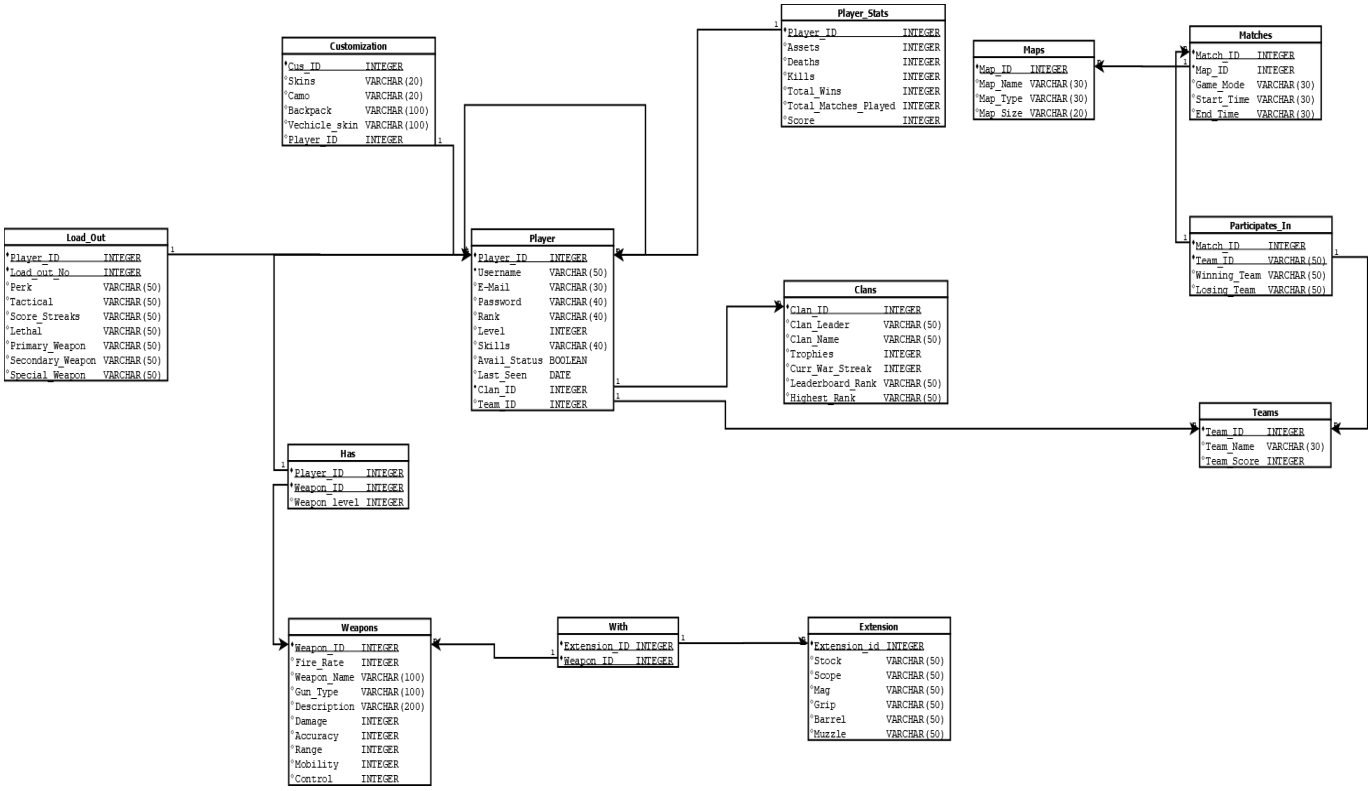
1. **Centralized Information Hub:** Create a centralized repository for all relevant information related to Call of Duty, including player statistics, game maps, weapons, and strategies.
2. **Enhanced Gameplay Experience:** Provide players with easy access to comprehensive data that can help them improve their gameplay, make informed decisions, and enhance their overall gaming experience.
3. **Accessibility and Usability:** Ensure that the database is user-friendly and accessible to players of all skill levels, making it easy for both beginners and experienced gamers to navigate and utilize the wealth of information available.

By achieving these goals, the Call of Duty database project aims to serve as an invaluable resource for the gaming community, empowering players to enhance their skills, connect with fellow gamers, and enjoy the game to its fullest potential.

ER Diagram


End_Time
Start_Time
Match_ID
Game_Mode
Matches
N
Played on
1
Maps
Map_ID
Map_Name
Map_Type
Map_Size
Losing_Team
Winning_Team
N
Participates in
M
Team_ID
Team_Name
Teams
Team_Score
1
Part Of
N
M
Friend_Of
M
user_name
Grip
MAG
scope
Barrel
stock
Muzzel
Extension
Extension_id
N
with
M
Control
Weapon_ID
Fire_Rate
Mobility
Weapons
Range
Name
Accuracy
Damage
Description
Type
Level
N
has
M
Availability_Status
Player
Skills
E-Mail
Level
Rank
Passwprd
Last_Seen
Player_ID
N
Member of
1
Clans
Clan_ID
Clan_Leader
Clan_Name
Description
Trophies
Current_War_Streak
Highest_Rank
Leader_Board_Rank
Score
Assets
Death
Total_Matches_Played
Player_Stats
Total_Wins
Kills
1
Statistics
1
1
has
N
Load_out
Primary_Weapon
Sec_Weapon
Special_Weapon
Lethal
Load_Out_No.
Perk
Tactical
Score_Streaks
Camo
Customization_ID
Bag_pack
Customization
vehical_skin
Skin
N
unlocked
1

Relational Schema



Tables

Player Information

The "Player" table in the game's homepage encompasses essential information about each player. It includes a unique Player ID assigned to every user, ensuring individual distinction within the system. Each player selects a Username for identification purposes, and their Email is stored for account management. A secure Password is also stored to ensure account security. Additionally, the table records the player's Rank, reflecting their standing within the game's community, and their current Level of progression. Preferences such as Preferred Game Mode and Preferred Map are noted to tailor the gaming experience to the player's liking. The table also accounts for the player's Skills, showcasing their abilities or expertise within the game. Finally, Availability Status indicates whether a player is online or offline, facilitating real-time interaction and gameplay.

Clan

The "Clan" table contains crucial data regarding player affiliations within clans. It includes attributes such as unique Clan ID, Clan Leader, Clan Name, Trophies, Wins, Current War Streak, Leaderboard Rank, and Highest Rank. These details offer insights into clan performance, achievements, and competitive standing within the game's ecosystem.

Matches and Maps

The "Matches" and "Maps" tables hold key information essential for tracking gameplay and map details within the game. In the "Matches" table, data is recorded, including a unique Map ID, Start Time of the match, End Time of the match, as well as the Losing Team and Winning Team. This allows for precise tracking of match outcomes and performance metrics.

On the other hand, the "Maps" table stores vital attributes such as the unique Map ID, Map Name, Map Type, and Map Size. This information provides comprehensive insight into the diverse range of maps available within the game. The Map Name offers clear identification, while the Map Type categorizes maps based on their gameplay features (e.g., multiplayer, campaign, etc.). Additionally, Map Size indicates the dimensions and scale of each map, influencing gameplay dynamics and strategies.

Team

The "Team" table contains key details such as Team Name, Team Score, and a unique Team ID. This information provides a concise overview of each team's identity and performance within the game.

Weapons and Extensions

The weapon table in the Call of Duty database catalogs crucial attributes for each firearm, including Fire Rate, Damage, Accuracy, Range, Mobility, and Control. These details aid players in selecting the most effective weapons for their strategies. Additionally, the table's extensions, such as Muzzle, Barrel, Grip, Scope, Stock, and Extension_ID, offer customization options to fine-tune weapons for specific combat situations. This comprehensive resource empowers players to optimize their loadouts and excel in-game.

Load Out

The "Loadout" table serves as a repository for crucial information regarding the weapons and equipment utilized by players during the game. It includes attributes such as the player's primary weapon, secondary weapon, special weapon, lethal equipment, scorestreaks, tactical gear, perks, and loadout number. Each entry in the table represents a specific loadout chosen by a player for use in the game, allowing for detailed analysis of weapon preferences, tactical choices, and gameplay strategies. This data provides valuable insights into player preferences, playstyle variations, and the effectiveness of different loadout combinations in achieving success within the game.

Player Statistics

The Player_Stats table encompasses key statistics such as the player's overall score, total matches played, total wins, kills, deaths, and assets acquired throughout their gameplay.By providing insights into factors like win-loss ratio, kill-death ratio, and overall efficiency, the Player_Stats table enables players to gauge their performance objectively and identify areas for improvement.

1) Player

R(Player_ID,Username,E-Mail,Password,Rank,Level,Skills,Avail_Status,Last_seen,Clan_ID,Team_ID)

Keys: Player_ID

Minimal FD:

Player_ID → Username
Player_ID → E-Mail
Player_ID → Password
Player_ID → Rank
Player_ID → Level
Player_ID → Skills
Player_ID → Avail_Status
Player_ID → Last_Seen
Player_ID → Clan_ID
Player_ID → Team_ID

(Player_ID)+ = R(Player_ID, Username, E-mail, Password, Rank, Level, SKills, Avail_status, Last_Seen, Clan_ID, Team_ID)

Hence, Player_ID is the key

BCNF Proof :

Each minimal functional dependency mentioned indicates that Player_ID is a candidate key, thereby confirming that the relation meets the criteria for BCNF.

2) Clans

R(Clan_ID, Clan_Leader, Clan_Name, Trophies, Curr_War_Streak, Leaderboard_Rank, HIghest_Rank)

Keys: Clan_ID

Minimal FD:

Clan_ID → Clan_Leader
Clan_ID → Clan_Name
Clan_ID → Trophies
Clan_ID → Curr_War_Streak
Clan_ID → Leaderboard_Rank
Clan_ID → Highest_Rank

(Clan_ID)+ = R(Clan_ID, Clan_Leader, Clan_Name, Trophies, Curr_War_Streak, Leaderboard_Rank, Highest_Rank)

Hence, Clan_ID is the key

BCNF Proof :

Each minimal functional dependency mentioned indicates that Clan_ID is a candidate key, thereby confirming that the relation meets the criteria for BCNF.

3) Customization

R(Cus_ID, Skins, Camo, Backpack, Vehicle_skin, Player_ID)

Keys: Cus_ID

Minimal FD:

Cus_ID → Skins
Cus_ID → Camo
Cus_ID → Backpack
Cus_ID → Vehicle_Skin
Cus_ID → Player_ID

(Cus_ID)+ = R(Cus_ID, Skins, Camo, Backpack, Vehicle_Skin, Player_ID)

Hence, Cus_ID is the key

BCNF Proof :

Each minimal functional dependency mentioned indicates that Cus_ID is a candidate key, thereby confirming that the relation meets the criteria for BCNF.

4) Player_Stats

R(Player_ID, Assets, Deaths, Kills, Total_Wins, Total_Matches_Played, Score)

Keys: Player_ID

Minimal FD:

Player_ID → Assets
Player_ID → Deaths
Player_ID → Kills
Player_ID → Total_Wins
Player_ID → Total_Matches_Played
Player_ID → Score

(Player_ID)+ = R(Player_ID, Assets, Deaths, Kills, Total_Wins,Total_Matches_Played, Score)

Hence, Player_ID is the key

BCNF Proof :

Each minimal functional dependency mentioned indicates that Player_ID is a candidate key, thereby confirming that the relation meets the criteria for BCNF.

5) Maps

R(Map_ID, Map_Name, Map_Type, Map_Size)

Keys: Map_ID

Minimal FD:

Map_ID → Map_Name
Map_ID → Map_Type
Map_ID →Map_Size

(Map_ID)+ = R(Map_ID, Map_Name, Map_Type, Map_Size)

Hence, Map_ID is the key

BCNF Proof :

Each minimal functional dependency mentioned indicates that Map_ID is a candidate key, thereby confirming that the relation meets the criteria for BCNF.

6) Matches

R(Match_ID,Map_ID, Game_Mode, Start_Time, End_Time)

Keys: Match_ID

Minimal FD:

Match_ID → Map_ID
Match_ID → Game_Mode
Match_ID → Start_Time
Match_ID → End_Time

{Match_ID}+ = R(Match_ID,Map_ID,Game_Mode, Start_Time, End_Time)

Hence, Match_ID is the key

BCNF Proof :

Each minimal functional dependency mentioned indicates that Match_ID is a candidate key, thereby confirming that the relation meets the criteria for BCNF.

7) Teams

R(Team_ID, Team_Name, Team_Score)

Keys: Team_ID

Minimal FD:

Team_ID → Team_Name
Team_ID → Team_Score

{Team_ID}+ = R(Team_ID, Team_Name, Team_Score)

Hence, Team_ID is the key

BCNF Proof :

Each minimal functional dependency mentioned indicates that Team_ID is a candidate key, thereby confirming that the relation meets the criteria for BCNF.

8) Participates_In

R(Match_ID, Team_ID, Winning_Team, Losing_Team)

Keys: Match_ID,Team_ID

Minimal FD:

{Match_ID,Team_ID} → Winning_Team
{Match_ID,Team_ID} → Losing_Team

{Match_ID,Team_ID}+ = R(Match_ID, Team_ID, Winning_Team, Losing_Team)

Hence, {Match_ID,Team_ID} is the key.

BCNF Proof :

Each minimal functional dependency mentioned indicates that {Match_ID,Team_ID} is a candidate key, thereby confirming that the relation meets the criteria for BCNF.

9) Load_Out

R(Player_ID, Load_Out_No, Perk, Tactical, Score_Streaks, Lethal, Primary_Weapon, Secondary_Weapon, Special_Weapon)

Keys: Player_ID, Load_Out_No

Minimal FD:

{Player_ID, Load_Out_No} → Perk
{Player_ID, Load_Out_No} → Tactical
{Player_ID, Load_Out_No} → Score_Streaks
{Player_ID, Load_Out_No} → Lethal
{Player_ID, Load_Out_No} → Primary_Weapon
{Player_ID, Load_Out_No} → Secondary_Weapon
{Player_ID, Load_Out_No} → Special_Weapon

{Player_ID,Load_Out_No}+ = R(Player_ID,Load_Out_No,perk, Tactical, Score_Streaks, Lethal, Primary_Weapon, Secondary_Weapon, Special_Weapon)

Hence {Player_ID, Load_Out_No} is the key.

BCNF Proof :

Each minimal functional dependency mentioned indicates that {Player_ID, Load_Out_No} is a candidate key, thereby confirming that the relation meets the criteria for BCNF.

10) Has

R(Player_ID,Weapon_ID)

Keys: Player_ID,Weapon_ID

Minimal FD:
{Player_ID,Weapon_ID}+ = R(Player_ID,Weapon_ID) Hence, {Player_ID,Weapon_ID} s the key
BCNF Proof : Each minimal functional dependency mentioned indicates that {Player_ID,Weapon_ID} is a candidate key, thereby confirming that the relation meets the criteria for BCNF.

11) Weapons

R(Weapon_ID,Fire_Rate,Weapon_Level,Weapon_Name,Gun_Type,Description,Damage, Accuracy,Range ,Mobility, Control)

Keys: Weapon_ID

Minimal FD: Weapon_ID → Fire_Rate Weapon_ID → Weapon_Level Weapon_ID → Weapon_Name Weapon_ID → Gun_Type Weapon_ID → Description Weapon_ID → Damage Weapon_ID → Accuracy Weapon_ID → Range Weapon_ID → Mobility Weapon_ID → Control
{Weapon_ID}+ = R(Player_ID, Username, E-mail, Password, Rank, Level, SKills, Avail_status, Last_Seen, Clan_ID, Team_ID) Hence, Weapon_ID is the key
BCNF Proof : Each minimal functional dependency mentioned indicates that Weapon_ID is a candidate key, thereby confirming that the relation meets the criteria for BCNF.

12) With

R(Extension_ID,Weapon_ID)

Keys: Extension_ID, Weapon_ID

Minimal FD:
{Extension_ID,Weapon_ID}+ = R(Extension_ID, Weapon_ID) Hence,{Extension_ID,Weapon_ID} is the key
BCNF Proof : Each minimal functional dependency mentioned indicates that {Extension_ID,Weapon_ID} is a candidate key, thereby confirming that the relation meets the criteria for BCNF.

13) Extension

R(Extension_ID, Stock,Scope,MAG,Grip, Barrel, Muzzle)

Keys: Extension_ID

Minimal FD: Extension_ID → Stock Extension_ID → Scope Extension_ID → MAG Extension_ID → Grip Extension_ID → Barrel Extension_ID → Muzzle
(Extension_ID)+ = R(Extension_ID, Stock, Scope, MAG, Grip, Barrel, Muzzle) Hence, Extension_ID is the key
BCNF Proof : Each minimal functional dependency mentioned indicates that Extension_ID is a candidate key, thereby confirming that the relation meets the criteria for BCNF.