

Functional Dependencies:

USER(User_id, username, password, role, email, following, followers, favorite_characters, comment_id):

FD: User_id -> { username, password, role, email, following, followers, favorite_character, Comment_id }

FD: User_id, Email -> { username, password }

FD: Username -> { role, followers, following }

CHARACTER(character_id, character_name, game_series, range, fighter_style, Date_released, character_overall_rating(Derived)):

FD: { Character_id, Game_series } -> date_released

FD: Fighter_Style -> range

FD: Character_id -> { character_name, Game_series, Fighter_style, character_overall_rating }

Tier_List(Tier_list_id, user_id, ranking_list, character_id, comment_id):

FD: Tier_list_id -> { user_id, ranking_list, character_id, comment_id }

FD: { Tier_list_id, User_id } -> comment_id

FD: { Tier_list_id, User_id, Ranking_list } -> Character_id

OR

FD: { Tier_list_id, Character_id } -> ranking_list

Competitive_Player_Profile(Pro_id, Pro_fname, Pro_lname , Pro_stage_name,
Character_played, Upcoming_competition, Date_upcoming,
Name_upcoming, Name_history, Date_history,
History_competition, Win_lose_ratio):

FD: Pro_stage_name -> Pro_id

FD:Pro_id -> {Pro_fname, Pro_lname}

FD:Pro_id -> win_lose_ratio

FD:Pro_id -> Character_played

FD: {Pro_id , Competition_History} -> {Name_history, Date_History}

FD:{Pro_id , Upcoming_competition} -> {Name_upcoming, date_upcoming}

Air_smash(Character_id, up_air, forward_air, total_air_smash, neutral_air, down_air, back_air):

FD: Character_id -> up_air, forward_air, neutral_air, down_air, back_air

FD: Character_id, up_air, forward_air, neutral_air, down_air, back_air -> Total_air_smash

Ground_Smash(Character_id,up_smash, side_smash, down_smash, total_ground_smash,
neutral_smash):

Character_id -> up_smash, side_smash, down_smash, total_ground_smash, neutral_smash

FD: Character_id, up_smash, side_smash, down_smash, neutral_smash -> Total_ground_smash

Rating_smash(Character_id, ground_smash, Air_smash, total_smash):

FD:Character_id -> ground_smash, Air_smash

FD:ground_smash, Air_smash -> Total_smash

Rating_special(Character_id, Up_special, Side_special, Down_special, Neutral_special,
total_special):

FD:Character_id -> Up_special, Side_special, Down_special, Neutral_special

FD: Up_special, Side_special, Down_special, Neutral_special -> Total_Special

Rating(Character_id, total_smash, total_specials, Rating_special, rating_smash):

FD:Character_id -> total_smash, total_special

FD:Total_special -> Rating_special

FD:Total_smash -> rating_smash

Comments(Comment_id, user_id, comment_text):

FD: Comment_id, user_id -> comment_text

Normalization Forms:

CHARACTER(character_id, character_name, game_series, range, fighter_style, character_overall_rating)

1NF:

Character(Character_id, character_name, game_series, range, fighter_style, date_released, character_overall_rating)

2NF:

Character_id -> Character_name
Character_id -> fighter_style
Fighting_style -> range
Character_id -> Character_overall_rating
Character_id , Game_series -> Date_released

3NF:

Fighter_style -> range

BCNF:

Character_id , Character_name -> Character_id

Character(Character_id, Game_series, Fighter_style, Character_overall_rating)

CName(Character_id, Character_name)

Technique(Fighting_style, Range)

Released(Character_id, Game_series, Date_released)

USER(User_id, username, password, role, email, following, followers, favorite_character, comment_id)

1NF:

USER(User_id, username, password, role, email, following, followers, favorite_character, comment_id)

2NF:

User_id -> Favorite_character
User_id -> Comment_id
User_id, Email -> username, password
Username -> role, follower, following

3NF:

User_id, Email -> role, follower, following

BCNF:

User_id, Email -> User_id

User(User_id, Email, Favorite_character, comment_id)

Login(User_id, username, password)

Info(Username, User_ID, role, follower, following)

Tier_List(Tier_list_id, user_id, ranking_list, character_id, comment_id)

1NF:

Tier_List(Tier_list_id, user_id, ranking_list, character_id, comment_id)

2NF:

Tier_list_id, User_id -> Comment_id

Tier_list_id, User_id, Ranking_list -> Character_id

3NF:

No non-primary attribute is transitively dependent on primary key

BCNF:

Tier_list_id, User_id, Ranking_list -> Tier_list_id

TierList(Tier_list_id, User_id, Ranking_list)

Comment(Tier_list_id, User_id, Comment_id)

CharacterRank(Tier_list_id, User_id, Ranking_list, Character_id)

Competitive_Player_Profile(Pro_id, Pro_fname, Pro_lname, Pro_stage_name,

Character_played, Upcoming_competition, Date_upcoming,

Name_upcoming, Competition_history, Date_history,

Name_history, Win_lose_ratio):

1NF:

Competitive_player_profile(Pro_id, Pro_fname, Pro_lname, Pro_stage_name,
Character_played, Upcoming_competition, Date_upcoming,
Name_of_upcoming_competition, Competition_history, Date_history,
History_competition, Win_lose_ratio)

2NF:

Pro_id -> Pro_fname, Pro_lname

Pro_id -> Win_lose_ratio

Pro_id -> Character_played

Pro_id, Competition_History -> Name_history, Date_history

Pro_id, Upcoming_competition -> Name_upcoming, Date_upcoming

3NF:

No non-primary attribute is transitively dependent on primary key

BCNF:

Pro_id, Pro_stage_name -> Pro_id

Pro(Pro_id, Competition_History, Upcoming_competition)

ProInfo(Pro_id, Pro_fname, Pro_lname, Win_lose_ratio, Character_played)

History(Pro_id, Competition_History, Name_history, Date_history)

Upcoming(Pro_id, Upcoming_competition, Name_upcoming, Date_upcoming)

Air_smash(Character_id, up_air, forward_air, total_air_smash, neutral_air, down_air, back_air)

1NF:

Air_smash(Character_id, up_air, forward_air, total_air_smash, neutral_air, down_air, back_air)

2NF:

Character_id -> Up_air, forward_air, neutral_air, down_air, back_air

Character_id, Up_air, forward_air, neutral_air, down_air, back_air -> Total_air_smash

3NF:

No non-primary attribute is transitively dependent on primary key

BCNF:

Character_id is a candidate key because it has access to every attribute type

Air_Smash(Character_id, up_air, forward_air, neutral_air, down_air, back_air, Total_air_smash)

Ground_Smash(Character_id, up_smash, side_smash, down_smash, total_ground_smash, neutral_smash)

1NF:

Ground_Smash(Character_id, up_smash, side_smash, down_smash, total_ground_smash, neutral_smash)

2NF:

Character_id -> up_smash, side_smash, down_smash, total_ground_smash,
neutral_smash

Character_id, up_smash, side_smash, down_smash, total_ground_smash, neutral_smash
->Total_ground_smash

3NF:

No non-primary attribute is transitively dependent on primary key

BCNF:

Character_id is a candidate key because it has access to every attribute type

Ground_Smash(Character_id, up_smash, side_smash, down_smash, total_ground_smash,
Neutral_smash, Total_ground_smash)

Rating_smash(Character_id, ground_smash, Air_smash, total_smash)

1NF:

Rating_smash(Character_id, ground_smash, Air_smash, total_smash)

2NF:

Character_id -> ground_smash, Air_smash
Character_id, ground_smash, Air_smash -> Total_smash

3NF:

No non-primary attribute is transitively dependent on primary key

BCNF:

Character_id is a candidate key because it has access to every attribute type

Smash(Character_id, ground_smash, Air_smash, Total_Smash)

Rating_special(Character_id, Up_special, Side_special, Down_special, Neutral_special, total_special)

1NF:

Rating_special(Character_id, Up_special, Side_special, Down_special, Neutral_special, total_special)

2NF:

Character_id -> Up_special, Side_special, Down_special, Neutral_special
Character_id, Up_special, Side_special, Down_special, Neutral_special -> Total_special

3NF:

No non-primary attribute is transitively dependent on primary key

BCNF:

Character_id is a candidate key because it has access to every attribute type

Special(Character_id, Up_special, Side_special, Down_special, Neutral_special, total_special)

Rating(Character_id, total_smash, total_specials, Rating_special, rating_smash)

1NF:

Rating(Character_id, total_smash, total_specials, Rating_special, rating_smash)

2NF:

Character_id -> total_smash, total_specials

Character_id, Total_special -> Rating_special

Character_id, Total_smash -> Rating_smash

3NF:

No non-primary attribute is transitively dependent on primary key

BCNF:

Character_id is a candidate key because it has access to every attribute type

Rating(Character_id, Total_special, Total_smash)

Rating_spe(Character_id, Total_Special, Rating_special)

Rating_sma(Character_id, Total_smash, Rating_smash)

Comments(Comment_id, user_id, comment_text)

1NF:

Comments(Comment_id, user_id, comment_text)

2NF:

Comment_id, user_id -> comment_text

3NF:

No non-primary attribute is transitively dependent on primary key

BCNF:

Comment_id, user_id -> comment_id

CommentInfo(Comment_id, user_id)

CommentText(Comment_id, CommentTextInfo)