

TIPPR

Database Design & Entity Model

Version 1.0 | 2025-12-14

Översikt

Detta dokument beskriver den kompletta databasdesignen för Tippr-applikationen. Databasen är designad för PostgreSQL (via Supabase) och använder Entity Framework Core med Code-First approach.

Design Principer:

- **Normalisering:** 3NF (Third Normal Form) för att undvika redundans
- **UUID Primary Keys:** Använder Guid/UUID för alla primary keys
- **Timestamps:** CreatedAt och UpdatedAt på alla entities
- **Soft Deletes:** IsDeleted flag istället för faktisk borttagning (där relevant)
- **Foreign Keys:** All relationell integritet enforced på databas-nivå
- **Indexes:** På foreign keys och vanliga query-fält

Entity-Översikt

Entity	Beskrivning	Relationer
User	Användarinformation och profil	Supabase auth.users (1:1)
League	Liga/grupp för tippning	User (many-to-one för owner)
LeagueMember	Medlemskap i liga	User, League (many-to-many resolver)
LeagueSettings	Ligans konfiguration	League (one-to-one)
Tournament	Mästerskap (VM, EM, etc)	Standalone
Team	Landslag i turnering	Tournament (many-to-one)
Match	Matcher i turnering	Tournament, Team (many-to-one)
Prediction	Användarens matchtips	User, Match, League (many-to-one)
BonusQuestion	Bonusfrågor för turnering	Tournament (many-to-one)
BonusPrediction	Svar på bonusfrågor	User, BonusQuestion, League
LeagueStanding	Poängställning i liga	User, League (many-to-one)
ChatMessage	Meddelanden i liga-chat	User, League (many-to-one)
Notification	Användarnotifikationer	User (many-to-one)

Detaljerade Entity-Definitioner

1. User

Beskrivning: Representerar en användare i systemet. Länkas till Supabase auth.users.

Fält	Typ	Beskrivning	Constraints
Id	Guid (UUID)	Primary key, samma som auth.users.id	PK, NOT NULL
Username	string	Användarnamn (unikt)	UNIQUE, NOT NULL, MaxLength(50)
DisplayName	string	Visningsnamn	MaxLength(100)
Email	string	Email (från Supabase)	NOT NULL
AvatarUrl	string?	Profilbild URL	Nullable
Bio	string?	Användarens bio	MaxLength(500)
FavoriteTeamId	Guid?	Favoritlag	FK to Team, Nullable
CreatedAt	DateTime	Skapad datum	NOT NULL, Default: NOW()
UpdatedAt	DateTime	Uppdaterad datum	NOT NULL
LastLoginAt	DateTime?	Senaste inloggning	Nullable

Relationer:

- One-to-Many: User → LeagueMember
- One-to-Many: User → League (som owner)
- One-to-Many: User → Prediction
- One-to-Many: User → LeagueStanding

Indexes: Username (UNIQUE), Email

2. Tournament

Beskrivning: Representerar ett mästerskap (t.ex. Fotbolls-VM 2026).

Fält	Typ	Beskrivning	Constraints
Id	Guid	Primary key	PK, NOT NULL
Name	string	Namn (t.ex. "VM 2026")	NOT NULL, MaxLength(100)
Year	int	År	NOT NULL
Type	TournamentType	Typ (WorldCup, EuroCup, etc)	NOT NULL, Enum
StartDate	DateTime	Startdatum	NOT NULL
EndDate	DateTime	Slutdatum	NOT NULL
Country	string	Världsländ	MaxLength(100)
LogoUrl	string?	Turneringens logo	Nullable
IsActive	bool	Pågående/aktiv turnering	NOT NULL, Default: true
CreatedAt	DateTime	Skapad datum	NOT NULL

Relationer:

- One-to-Many: Tournament → Team
- One-to-Many: Tournament → Match
- One-to-Many: Tournament → BonusQuestion

Enums: TournamentType { WorldCup, EuroCup, ChampionsLeague, EuropaLeague }

3. Team

Beskrivning: Representerar ett landslag i en specifik turnering.

Fält	Typ	Beskrivning	Constraints
Id	Guid	Primary key	PK, NOT NULL
TournamentId	Guid	Vilken turnering	FK to Tournament, NOT NULL
Name	string	Lagets namn (t.ex. "Sverige")	NOT NULL, MaxLength(100)
Code	string	Landskod (t.ex. "SWE")	NOT NULL, MaxLength(3)
FlagUrl	string?	URL till flagga	Nullable
GroupName	string?	Grupp (t.ex. "A", "B")	MaxLength(5)
ApiFootballId	int?	ID från API-FOOTBALL	Nullable

Relationer:

- Many-to-One: Team → Tournament
- One-to-Many: Team → Match (som HomeTeam eller AwayTeam)

Indexes: TournamentId, Code

4. Match

Beskrivning: Representerar en match mellan två lag.

Fält	Typ	Beskrivning	Constraints
Id	Guid	Primary key	PK, NOT NULL
TournamentId	Guid	Vilken turnering	FK to Tournament, NOT NULL
HomeTeamId	Guid	Hemmalag	FK to Team, NOT NULL
AwayTeamId	Guid	Bortalag	FK to Team, NOT NULL
MatchDate	DateTime	När matchen spelas	NOT NULL
Stage	MatchStage	Stadie (Group, R16, etc)	NOT NULL, Enum
HomeScore	int?	Hemmalags mål	Nullable
AwayScore	int?	Bortalags mål	Nullable
Status	MatchStatus	Status (Scheduled, Live, etc)	NOT NULL, Enum
Venue	string?	Arena	MaxLength(200)
ApiFootballId	int?	ID från API-FOOTBALL	Nullable
UpdatedAt	DateTime	Senast uppdaterad	NOT NULL

Enums:

- MatchStage { Group, RoundOf16, QuarterFinal, SemiFinal, Final }
- MatchStatus { Scheduled, Live, FullTime, Postponed, Cancelled }

Indexes: TournamentId, MatchDate, Status

5. League

Beskrivning: En liga/grupp där användare tävlar tillsammans.

Fält	Typ	Beskrivning	Constraints
Id	Guid	Primary key	PK, NOT NULL
Name	string	Ligans namn	NOT NULL, MaxLength(100)
Description	string?	Beskrivning	MaxLength(500)
TournamentId	Guid	Vilken turnering	FK to Tournament, NOT NULL
OwnerId	Guid	Ligans skapare	FK to User, NOT NULL
InviteCode	string	Unik inbjudningskod	UNIQUE, NOT NULL, MaxLength(20)
IsPublic	bool	Publik eller privat	NOT NULL, Default: false
IsGlobal	bool	Global liga (alla kan delta)	NOT NULL, Default: false
MaxMembers	int?	Max antal medlemmar	Nullable
ImageUrl	string?	Ligans bild	Nullable
CreatedAt	DateTime	Skapad datum	NOT NULL
UpdatedAt	DateTime	Uppdaterad datum	NOT NULL

Relationer:

- Many-to-One: League → User (owner)
- Many-to-One: League → Tournament
- One-to-One: League → LeagueSettings
- One-to-Many: League → LeagueMember

Indexes: InviteCode (UNIQUE), TournamentId, OwnerId

6. LeagueSettings

Beskrivning: Konfigurerbara inställningar för en liga.

Fält	Typ	Beskrivning	Constraints
Id	Guid	Primary key	PK, NOT NULL
Leagueld	Guid	Tillhörande liga	FK to League, NOT NULL, UNIQUE
PredictionMode	PredictionMode	Tippningsläge	NOT NULL, Enum
DeadlineMinutes	int	Deadline innan match (min)	NOT NULL, Default: 60
PointsCorrectScore	int	Poäng för exakt resultat	NOT NULL, Default: 7
PointsCorrectOutcome	int	Poäng för rätt utkomst	NOT NULL, Default: 3
PointsCorrectGoals	int	Poäng per rätt antal mål	NOT NULL, Default: 2
PointsRoundOf16Team	int	Poäng per lag i åttondel	NOT NULL, Default: 2
PointsQuarterFinalTeam	int	Poäng per lag i kvart	NOT NULL, Default: 4
PointsSemiFinalTeam	int	Poäng per lag i semi	NOT NULL, Default: 6
PointsFinalTeam	int	Poäng per lag i final	NOT NULL, Default: 8
PointsTopScorer	int	Poäng för skyttekung	NOT NULL, Default: 20
PointsWinner	int	Poäng för vinnare	NOT NULL, Default: 20
PointsMostGoalsGroup	int	Flest mål grupp spel	NOT NULL, Default: 10
PointsMostConcededGroup	int	Flest insläppta grupp	NOT NULL, Default: 10
AllowLateEdits	bool	Tillåt ändringar efter deadline	NOT NULL, Default: false

Enums: PredictionMode { AllAtOnce, StageByStage, MatchByMatch }

7. LeagueMember

Beskrivning: Representerar medlemskap i en liga (many-to-many mellan User och League).

Fält	Typ	Beskrivning	Constraints
Id	Guid	Primary key	PK, NOT NULL
Leagueld	Guid	Ligan	FK to League, NOT NULL
UserId	Guid	Användaren	FK to User, NOT NULL
JoinedAt	DateTime	När användaren gick med	NOT NULL
IsAdmin	bool	Är administratör	NOT NULL, Default: false
IsMuted	bool	Mutad i chatten	NOT NULL, Default: false

Constraints: UNIQUE(Leagueld, UserId) - En användare kan bara vara med en gång per liga

Indexes: Leagueld, UserId, Composite (Leagueld, UserId)

8. Prediction

Beskrivning: En användares tips på en match i en specifik liga.

Fält	Typ	Beskrivning	Constraints
Id	Guid	Primary key	PK, NOT NULL
UserId	Guid	Vem som tippat	FK to User, NOT NULL
MatchId	Guid	Vilken match	FK to Match, NOT NULL
Leagueld	Guid	Vilken liga	FK to League, NOT NULL
HomeScore	int	Tippat resultat hemma	NOT NULL
AwayScore	int	Tippat resultat borta	NOT NULL
PointsEarned	int?	Poäng erhållna	Nullable (null = ej avgjord)
CreatedAt	DateTime	När tips skapades	NOT NULL
UpdatedAt	DateTime	När tips uppdaterades	NOT NULL

Constraints: UNIQUE(UserId, MatchId, Leagueld) - Ett tips per match per liga

Indexes: UserId, MatchId, Leagueld, Composite (UserId, Leagueld)

9. BonusQuestion

Beskrivning: Bonusfrågor för en turnering (skyttkung, vinnare, etc).

Fält	Typ	Beskrivning	Constraints
Id	Guid	Primary key	PK, NOT NULL
TournamentId	Guid	Vilken turnering	FK to Tournament, NOT NULL
QuestionType	BonusQuestionType	Typ av fråga	NOT NULL, Enum
Question	string	Frågetext	NOT NULL, MaxLength(200)
AnswerTeamId	Guid?	Rätt svar (lag)	FK to Team, Nullable
AnswerText	string?	Rätt svar (text)	Nullable, MaxLength(200)
IsResolved	bool	Är besvarad	NOT NULL, Default: false
Points	int	Poäng för rätt svar	NOT NULL

Enums: BonusQuestionType { Winner, TopScorer, MostGoalsGroup, MostConcededGroup, RoundOf16Teams, QuarterFinalTeams, SemiFinalTeams, FinalTeams }

10. BonusPrediction

Beskrivning: Användarens svar på en bonusfråga.

Fält	Typ	Beskrivning	Constraints
Id	Guid	Primary key	PK, NOT NULL
UserId	Guid	Vem som svarat	FK to User, NOT NULL
BonusQuestionId	Guid	Vilken fråga	FK to BonusQuestion, NOT NULL
Leagueld	Guid	Vilken liga	FK to League, NOT NULL
AnswerTeamId	Guid?	Svar (lag)	FK to Team, Nullable
AnswerText	string?	Svar (text)	Nullable, MaxLength(200)
PointsEarned	int?	Poäng erhållna	Nullable
CreatedAt	DateTime	När svar skapades	NOT NULL

Constraints: UNIQUE(UserId, BonusQuestionId, Leagueld)

11. LeagueStanding

Beskrivning: Poängställning för en användare i en liga (ligatabellen).

Fält	Typ	Beskrivning	Constraints
Id	Guid	Primary key	PK, NOT NULL
Leagueld	Guid	Vilken liga	FK to League, NOT NULL
UserId	Guid	Vilken användare	FK to User, NOT NULL
TotalPoints	int	Totala poäng	NOT NULL, Default: 0
MatchPoints	int	Poäng från matcher	NOT NULL, Default: 0
BonusPoints	int	Poäng från bonusfrågor	NOT NULL, Default: 0
Rank	int	Placering i ligan	NOT NULL
PreviousRank	int?	Föregående placering	Nullable
UpdatedAt	DateTime	Senast uppdaterad	NOT NULL

Constraints: UNIQUE(Leagueld, UserId)

Indexes: Leagueld, Composite (Leagueld, Rank) för snabb sortering

Note: Denna tabell uppdateras efter varje match och används för caching av ligatabellen.

12. ChatMessage

Beskrivning: Meddelanden i liga-chatten.

Fält	Typ	Beskrivning	Constraints
Id	Guid	Primary key	PK, NOT NULL
Leagueld	Guid	Vilken liga	FK to League, NOT NULL
UserId	Guid	Vem som skrev	FK to User, NOT NULL
Message	string	Meddelandetext	NOT NULL, MaxLength(1000)
CreatedAt	DateTime	När meddelandet skickades	NOT NULL
IsEdited	bool	Har redigerats	NOT NULL, Default: false
EditedAt	DateTime?	När det redigerades	Nullable
IsDeleted	bool	Är borttaget (soft delete)	NOT NULL, Default: false

Indexes: Leagueld, CreatedAt för paginering och kronologisk sortering

13. Notification

Beskrivning: Notifikationer till användare.

Fält	Typ	Beskrivning	Constraints
Id	Guid	Primary key	PK, NOT NULL
UserId	Guid	Mottagare	FK to User, NOT NULL
Type	NotificationType	Typ av notifikation	NOT NULL, Enum
Title	string	Rubrik	NOT NULL, MaxLength(200)
Message	string	Meddelande	NOT NULL, MaxLength(500)
RelatedEntityId	Guid?	Relaterad entity (Match, League)	Nullable
IsRead	bool	Är läst	NOT NULL, Default: false
CreatedAt	DateTime	När notifikation skapades	NOT NULL
ReadAt	DateTime?	När den lästes	Nullable

Enums: NotificationType { MatchResult, DeadlineReminder, RankChange, NewChatMessage, LeagueInvite, MatchStarting }

Indexes: UserId, IsRead, CreatedAt

Entity Relationship Diagram

Huvudrelationer:

- **User ↔ auth.users:** 1:1 relation via UUID
- **User → League:** En användare kan äga många ligor (OwnerId)
- **User ↔ League:** Many-to-Many via LeagueMember (kan vara med i många ligor)
- **League → Tournament:** Many-to-One (en liga hör till en turnering)
- **League ↔ LeagueSettings:** 1:1 (en liga har en settingskonfiguration)
- **Tournament → Team:** One-to-Many (en turnering har många lag)
- **Tournament → Match:** One-to-Many (en turnering har många matcher)
- **Match → Team:** Many-to-One × 2 (HomeTeam och AwayTeam)
- **User → Prediction:** One-to-Many (en användare kan ha många tips)
- **Match → Prediction:** One-to-Many (en match kan ha många tips)
- **League → Prediction:** One-to-Many (tips är per liga)
- **Tournament → BonusQuestion:** One-to-Many
- **BonusQuestion → BonusPrediction:** One-to-Many
- **League → LeagueStanding:** One-to-Many (en post per användare)
- **League → ChatMessage:** One-to-Many
- **User → Notification:** One-to-Many

Vanliga Queries & Index Strategi

1. Hämta ligatabell sorterad på poäng:

```
SELECT * FROM LeagueStanding WHERE LeagueId = @leagueId ORDER BY TotalPoints DESC,  
UpdatedAt ASC
```

Index: Composite index på (LeagueId, TotalPoints)

2. Hämta användarens alla tips för en liga:

```
SELECT p.*, m.* FROM Prediction p JOIN Match m ON p.MatchId = m.Id WHERE p.UserId =  
@userId AND p.LeagueId = @leagueId ORDER BY m.MatchDate
```

Index: Composite index på (UserId, LeagueId)

3. Hämta kommande matcher för en turnering:

```
SELECT * FROM Match WHERE TournamentId = @tournamentId AND Status = 'Scheduled' AND  
MatchDate > NOW() ORDER BY MatchDate
```

Index: Composite index på (TournamentId, Status, MatchDate)

4. Räkna ut poäng efter en match (batch query):

```
SELECT p.*, m.HomeScore, m.AwayScore, ls.* FROM Prediction p JOIN Match m ON  
p.MatchId = m.Id JOIN League l ON p.LeagueId = l.Id JOIN LeagueSettings ls ON l.Id =  
ls.LeagueId WHERE p.MatchId = @matchId
```

Index: MatchId på Prediction

Migration Strategi

Steg-för-steg process:

1. **Skapa Domain Entities** i .NET projektet (Domain Layer)
2. **Konfigurera DbContext** med FluentAPI för relationships
3. **Skapa Initial Migration:** dotnet ef migrations add InitialCreate
4. **Granska migration:** Kontrollera genererad SQL
5. **Applicera till Supabase:** dotnet ef database update
6. **Seed Data:** Skapa tournament, teams, matches (från API-FOOTBALL)
7. **Skapa Global League:** Default liga som alla kan gå med i

Connection String för Supabase:

```
Host=db.xxxxx.supabase.co;Database=postgres;Username=postgres;Password=your-password;SSL Mode=Require
```

Viktigt att tänka på:

- **Supabase använder 'public' schema** - inga ändringar behövs
- **Använd UTC timestamps** - Supabase kör PostgreSQL i UTC
- **UUID Generation:** Använd C# Guid.NewGuid() eller PostgreSQL gen_random_uuid()
- **Migrations i version control:** Commita alltid migrations till Git

Performance & Optimering

1. Indexering (redan nämnt ovan):

- Foreign keys: Automatisk index på alla FK
- Composite indexes: För vanliga multi-column queries
- UNIQUE constraints: Automatisk index

2. Caching Strategi:

- **LeagueStanding**: Materialiserad vy, uppdateras efter varje match
- **Match data**: Cacha i .NET IMemoryCache (5-15 min)
- **Team/Tournament**: Långvarig cache (ävändras sällan)

3. Query Optimering:

- Använd **AsNoTracking()** för read-only queries
- Använd **Select** för att bara hämta nödvändiga kolumner
- Använd **Include** istället för lazy loading
- Batch updates för LeagueStanding efter matcher

4. Skalning:

- **Read Replicas**: Supabase Pro har read replicas för read-heavy queries
- **Partitionering**: Överväg partitionering av Prediction och ChatMessage vid >1M rows
- **Archiving**: Flytta gamla turneringar till arkiv-tabeller

Sammanfattning

Denna databasdesign ger en solid grund för Tippr-applikationen med fokus på skalbarhet, prestanda och flexibilitet. Designen är normaliserad för att undvika redundans samtidigt som LeagueStanding-tabellen fungerar som en materialiserad vy för snabb access till ligatabeller.

Nästa steg:

1. Skapa C# entity-klasser baserat på denna design
2. Konfigurera EF Core DbContext med FluentAPI
3. Generera och applicera migrations till Supabase
4. Implementera Supabase Auth integration
5. Skapa seed data för första turneringen (VM 2026)

Dokument genererat: 2025-12-14 19:40
Version 1.0 | Tippr Database Design