

Disclaimer | De lay-out van de PDF-versie voor dit document kan verschillen met de markdown versie, voor een accurate weergave zie markdown bestand in bitbucket.

Technisch ontwerp

- Niet-functionele eisen
- Deployment diagram
 - Server unknown
 - Client
 - Transport Script
 - MongoDB
 - SQL Server
- PDM
- Toelichting Domeinen
- Ontwerpkeuzes
 - Stored procedures
 - MATCH id
 - EVENT id
 - Events
 - * Losse tabel voor iedere event
 - * Één event-tabel, children voor complexere events
 - * NoSQL
 - * Keuze
 - ADD_NEW_EVENT_TYPE
 - Moment van triggeren TRG_CHECK_PLAYER_COUNT
 - * Alternatief
 - Match_type
 - * Losse tabellen
 - * Één tabel
- Constraints
 - Primary key constraints
 - * Person
 - * Match
 - * Club
 - * City
 - * Country
 - * Competition
 - * Season
 - * Round
 - * Matchday
 - * Event
 - Integrity rules
- Toelichting export MSSQL naar MongoDB
 - Input
 - Output
- Stored Procedures
 - Koppeling stored procedures met use cases
 - Koppeling stored procedures met views

Niet-functionele eisen

Niet-functionele eisen	Categorieën
Data-analisten van NUTMEG lezen alleen data uit vanuit de MongoDB staging area, zo wordt de load op de SQL Server verlicht	Performance

Niet-functionele eisen	Categorieën
Administrators van NUTMEG hebben volledige CRUD rechten op zowel de SQL Server als MongoDB database	Security
Data-analisten hebben alleen lees rechten en lezen data exclusief uit vanuit de staging area	Security
De data in de MSSQL database wordt alleen in het Nederlands opgeslagen	Localization
Tabelnamen, constraints, triggers, views en procedures in SQL Server worden in het Engels geschreven	Localization
Data-analisten kunnen via de MongoDB shell data uitlezen	Usability
Iedere dag dient er om 02:00 een kopie van de SQL Server database overgezet te worden naar de MongoDB staging area	Reliability
De gegevens moeten bereikbaar zijn via een staging area in JSON formaat.	Usability

Deployment diagram

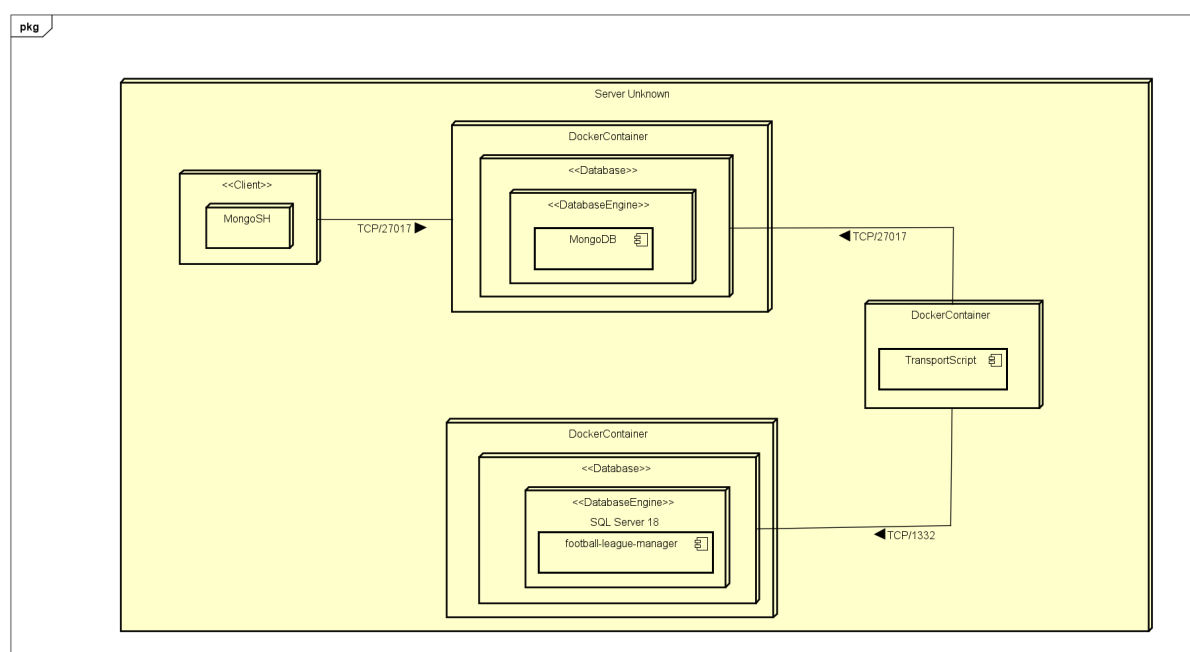


Figure 1: Deployment diagram.

Het bovenstaande diagram geldt alleen voor de ontwikkelomgeving.

Server unknown

In het bovenstaande diagram is **Server unknown** alles wat draait op de lokale computer van de ontwikkelaars.

Client

De client heeft een directe verbinding met de **MongoDB node** dit is de staging area die gebruikt wordt door data-analisten. Data-analisten krijgen niet direct toegang tot de **SQL Server node**, dit mogen alleen de admins.

Transport Script

Binnen deze container draaien alle scripts die van belang zijn voor het overzetten van de SQL Server JSON naar MongoDB.

MongoDB

De client interact met de MongoDB staging area, dit is ook hoe de uiteindelijke omgeving eruit komt te zien. De client is dus niet direct in verbinding met de SQL Server maar eerst met de staging area (MongoDB). Tijdens de constructie fase van het project zal ook direct verbinding gemaakt worden met de SQL Server, echter is dit in de uiteindelijke versie van de applicatie niet het geval (dan hebben alleen de admins direct toegang tot de SQL Server).

SQL Server

Binnen de SQL Server container draait een Microsoft SQL Server 2018 Server image. Vanaf deze node wordt alle data geleverd die gebruikt kan worden in de staging area. Deze node staat niet per direct in verbinding met de MongoDB node. Dit heeft als reden dat alle data eerst omgezet moet worden naar JSON die vervolgens MongoDB weer kan gebruiken om data mee te importeren.

PDM

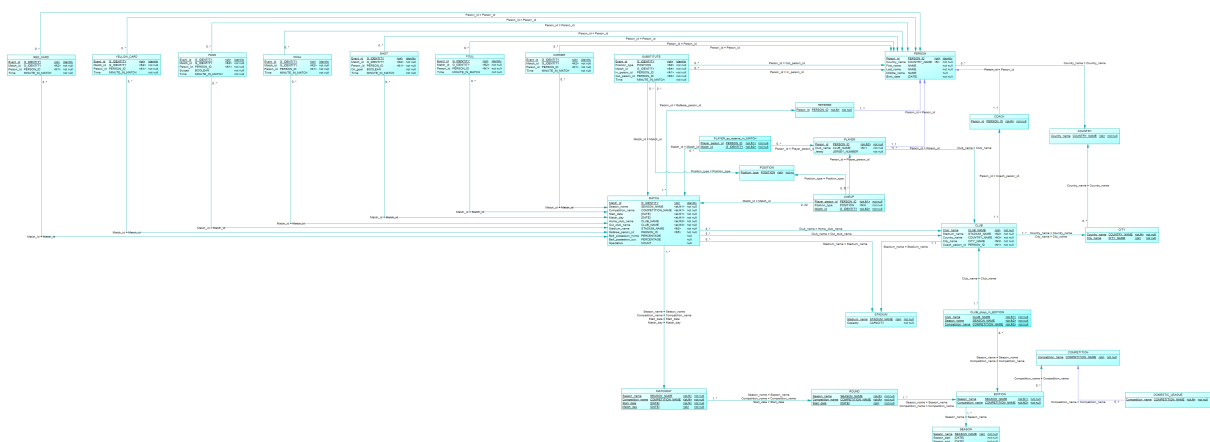


Figure 2: Het PDM.

Toelichting Domeinen

Domein	Datatype	Toelichting
PERSON_ID	bigint	Automatisch gegenereerd ID
NAME	varchar(128)	
DATE	Date	
CLUB_NAME	varchar(128)	
CITY_NAME	varchar(128)	
COUNTRY_NAME	varchar(128)	
STADIUM_NAME	varchar(60)	
COMPETITON_NAME	varchar(128)	
CAPACITY	bigint	
PERCENTAGE	numeric(5,2)	(0-100)
COUNT	bigint	
BOOLEAN	bit	
EVENTY_TYPE	varchar(128)	(Red card, Yellow card, Pass, Goal, Shot, Foul, Corner, Substitute)
MINUTE IN	numeric(6,3)	
MATCH		
JERSEY_NUMBER	numeric(2,0)	(1-99)
POSITION_TYPE	varchar(128)	(Aanvaller, Middenvelder, Verdediger, Keeper)
SEASON_NAME	char(5)	Altijd twee getallen met een slash. (bijv.22/21)

Ontwerpkeuzes

Stored procedures

Met het gebruik van stored procedures wordt de functionaliteit van de database geabstraheerd. De stored procedures zijn herbruikbaar en toegevoegde functionaliteit is overzichtelijk terug te vinden. Dit maakt het systeem beter uitbreidbaar en het verlaagd de koppeling tussen de database en de back-end. Verder wordt het execution plan van een stored procedure opgeslagen in de MSSQL Server-omgeving. Acties die vaak herhaald worden kunnen hierdoor sneller uitgevoerd worden.

MATCH id

Door de afhankelijkheden van match zou de originele primary key maar liefst 6 kolommen lang zijn. Event, position en Player_as_Reserve zouden daarnaast ook afhankelijk zijn van MATCH waardoor deze tabellen ook allemaal de 6 kolommen lange primary key zouden bevatten. Om deze reden is ervoor gekozen om match van een ID te voorzien. De originele identifier staat wel als alternative key in de database opgeslagen, zodat verzekerd is dat deze waarde ook daadwerkelijk uniek blijft.

EVENT id

Het is niet mogelijk om met de bestaande kolommen een primary key te maken voor EVENT. Hoewel de combinatie van minuut, persoon en type-overtreding op het eerste oog een kandidaat lijkt, gaat dit niet op aangezien het mogelijk is om meerdere overtredingen tegelijkertijd te maken (p-29 <https://www.knvb.nl/downloads/bestand/4841/spelregels-veldvoetbal-2021-22>). Omdat het natuurlijk wel mogelijk moet zijn naar iedere unieke EVENT te refereren, is ervoor gekozen de EVENTS van een id te voorzien.

Events

Het vastleggen van events kan op veel verschillende manieren. Om te bepalen wat de beste optie is moet rekening gehouden worden met vele aspecten, bijvoorbeeld: - Uitbreidbaarheid. Hoe makkelijk is het om events toe te voegen? - Hoe complex zijn joins/reads voor het genereren van de top-lijsten? - Hoe overzichtelijk en onderhoudbaar is de implementatie?

Losse tabel voor iedere event

Iedere event krijgt zijn eigen tabel met daarin de informatie die voor dat event belangrijk is.

Voordelen:

- Toplijsten generen kan voor iedere event met een simpele `COUNT(*) FROM <EVENT>`, dit houdt de code overzichtelijk en onderhoudbaar.

Nadelen:

- Voor het aanmaken van nieuwe events moeten er nieuwe tabellen aangemaakt worden, dit kan eventueel foutgevoelig zijn.

Één event-tabel, children voor complexere events

Simpele events (waarbij alleen het tijdstip in de wedstrijd en 1 persoon vereist zijn) komen in een parent tabel. Complexere events zullen een child van EVENT worden, ze erfen de basisattributen over en breiden deze uit.

Voordelen:

- Het toevoegen van simpele events kan makkelijk door admins zelf.

Nadelen:

- Minder overzichtelijk, sommige events zullen in de EVENT tabel staan terwijl andere EVENTS weer hun eigen tabel krijgen.

NoSQL

Events worden bijgehouden in een NoSQL database die geen moeite hebt verschillende aantallen attributen, bijvoorbeeld MongoDB.

Voordelen:

- Geen problemen met verschillende toegevoegde/ontbrekende attributen.

Nadelen:

- Lastiger om goed te onderhouden, gegevens staan in verschillende databases die vervolgens in de staging area weer gecombineerd zullen moeten worden.

Keuze

Hoewel het toevoegen van events op het eerste oog makkelijker lijkt bij de tweede optie, zullen de events die gebruikers eventueel zelf willen toevoegen waarschijnlijk geen simpele events zijn.

Verder is het toevoegen van niet-simpele events bij zowel optie 1 als 2 lastiger. Optie 3 maakt het onderhoud van het systeem veel lastiger en lijkt daarom ook geen goede optie.

Optie 1 is uiteindelijk geïmplementeerd aangezien het de meest overzichtelijke optie is en bijna zo goed uitbreidbaar is als optie 2.

ADD_NEW_EVENT_TYPE

Met deze procedure kunnen admins nieuwe events toevoegen zonder dat zij kennis hoeven te hebben van alle triggers, constraints en relaties die aan events zijn toe gewezen. Alle nodige integriteitsregels worden in de stored procedures afgehandeld.

Moment van triggeren TRG_CHECK_PLAYER_COUNT

De trigger checkt op inserts en updates, op deze manier worden er bij het toevoegen van spelers ook gekeken naar de huidige hoeveelheid spelers in een club binnen een match. Zo kunnen er per club minimaal 7 spelers en maximaal 11 spelers meedoen aan een wedstrijd. Dit geldt voor zowel de thuis als uit club (dus minimaal 14 en maximaal 22 in totaal per wedstrijd voor beide clubs).

Alternatief

Er zou ook gebruik gemaakt kunnen worden van een cronjob, zo kan er bij de start van een match gekeken worden naar het aantal spelers in een club die meedoen. Dit lijkt erg op onze huidige oplossing, echter wordt hier pas het aantal spelers afgevangen bij de start van een wedstrijd, zo zou er dus nog wel foutieve data in de database kunnen worden geïnsert; maar niet foutieve matches worden gestart.

Match_type

Binnen het systeem bestaan twee soorten matches, matches in een knockout toernooi en matches in een nationale competitie. Aangezien deze twee soorten matches verschillende relaties vereisen moet er goed nagedacht worden over de implementatie de matches.

Hiervoor zijn de volgende oplossingen bedacht.

Losse tabellen

Een optie is om voor zowel MATCH als KO_MATCH een losse tabel aan te maken. Met deze implementatie hoeft er niks aangepast te worden aan de huidige MATCH tabel

Echter loop je dan wel tegen een probleem aan met de events. De foreign keys zullen vanuit event naar twee verschillende tabellen moeten verwijzen, dat kan natuurlijk niet. Dit kan opgelost worden met een koppeltabel.

Vervolgens zullen de triggers van MATCH nog wel aangepast moeten worden en ook zullen er triggers aan KO-MATCH toegevoegd moeten worden.

Één tabel

Wanneer je ervoor kiest om alle matches in één tabel te stoppen hoef je de triggers van MATCH niet aan te passen. KO-matches zullen foreign keys krijgen naar de voorgaande wedstrijden. Reguliere matches zullen een nutteloze verwijzingen naar zichzelf krijgen aangezien de foreign key niet leeg kan zijn.

Constraints

Primary key constraints

Person

PI: Person ID

Match

PI: Club_name + club_name + match_day + start_date + end_date + competition_name + season_name

Club

PI: Club_name

City

PI: City_name

Country

PI: Country_name

Competition

PI: Competition_name

Season

PI: Season_name

Round

PI: Start_date + end_date

Matchday

PI: Match_day + start_date + end_date + competition_name + season_name

Event

PI: Time + club_name + club_name + match_day + start_date + end_date + competition_name + season_name

Integrity rules

				Komt overeen met
Nummer	Naam	Tabel	Omschrijving	
IR1	TRG_CHECKPOINT	PERSON	Er zijn minimaal 7 en maximaal 11 spelers per club opgesteld wanneer een wedstrijd start	C1 & BR12

Num	Naam	Tabel	Omschrijving	Komt overeen met
IR5	CHK_VALID_PLAYER	PLAYER	Een rugnummer van een speler moet hoger zijn dan 0 (mag niet 0 zijn) en mag niet hoger zijn dan 99 (mag wel 99 zijn)	C5 & BR17
IR6	CHK_VALID_ROUND_START_DATE	ROUND	De startdatum van een speelronde ligt binnen de start- en einddatum van het bijbehorende seizoen	C6 & BR4
IR7	TRG_CHECKMATCHDAYSTARTDATE	MATCHDAY	Een speeldag moet voor de startdatum van een opvolgende speelronde zijn, maar hetzelfde of na de startdatum van de gekoppelde speelronde	C7 & BR5
IR8	TRG_CHECKSHOT_PLAYER	SHOT & PLAYER	Een wedstrijd zijn gekoppeld in een wedstrijd; zijn van één van de twee clubs die aan de wedstrijd meedoen;	C8 & BR8
IR9	TRG_CHECKMATCH_IN_EDITION	MATCH	Een club mag alleen meedoen aan een wedstrijd als ze ook aan de bijbehorende editie meedoen	C9 & BR9
IR10	TRG_PERSONYELLOWCARD & RED_CARD	PERSON	Coaches en spelers mogen een rode of gele kaart krijgen	C10 & BR10
IR11	TRG_PERSONPASS, KICK, SHOT, FOUL, CORNER & SUBSTITUTE	PERSON	Alleen voor spelende spelers in een wedstrijd worden het aantal schoten, het aantal passes, de wissels, de overtredingen, de corners en of de persoon heeft gescoord bijgehouden	C11 & BR11
IR12	CHK_PERSON_MINOR_AGE	PERSON	Een persoon mag niet jonger zijn dan 15 jaar	C12 & BR20
IR13	TRG_VALID_MATCH_OF_SPECTATORS	MATCH	De toeschouwers bij een wedstrijd mag niet groter zijn dan de capaciteit van het stadion waar de wedstrijd wordt gehouden	C13 & BR21
IR14	CHK_VALID_MINUTE_IN_MATCH	MINUTE	De minuut in een wedstrijd mag niet negatief zijn	C14 & BR22
IR15	TRG_PLAYER_COACH, GOALKEEPER & REFEREE	PLAYER	Een persoon moet een speler, coach of scheidsrechter zijn	CDM, PDM & BR24
IR16	TRG_NO_UPLOADS_CURRENT_SEASON, COMPETITION & CLUB_plays_in_EDITION	CLUB	Van een editie mogen alleen de selecties van clubs en de speeldata van wedstrijden aangepast worden	C15 & BR1
IR17	TRG_MAX_16CLUBS_KNOCKOUT	CLUB	Knockout toernooien hebben 16 deelnemende clubs	

Toelichting export MSSQL naar MongoDB

Voor het omzetten van data in MSSQL Server naar MongoDB wordt gebruik gemaakt van een Python script.

De reden voor het gebruik van Python is omdat het snel is om code mee te schrijven en de code leesbaar is voor iedereen in de projectgroep. (Ook al is niet iedereen vaardig in het schrijven van Python code).

Hieronder staat toegelicht hoe dit script in elkaar zit:

Het script begint met een timer van 15 seconden, de reden hiervoor is omdat MSSQL en MongoDB opgestart moeten zijn voordat ermee verbonden kan worden. Ook moet MSSQL een (groot) aantal create-

en insert scripts uitvoeren, waar ook op gewacht moet worden.

Wanneer de verbindingen met de databases zijn gemaakt, wordt aan MSSQL opgevraagd welke tabellen en views er in de database staan. Vervolgens wordt alle data als JSON uit deze tabellen en views getrokken en wordt dit opgeslagen.

Zodra dit is gebeurd, wordt er over alle opgehaalde tabellen en views heen geloopt en worden deze als collections in MongoDB geplaatst.

Input

```
SELECT TOP 10 ID, FIRST_NAME, EMAIL, GENDER, IP_ADDRESS, A_BIG_NUMBER FROM MOCK_DATA;
```

id	first_name	email
1	Wells	wbraddick0@yellowbook.com
2	Phaidra	pdumbellow1@twitter.com
3	Taffy	tsherrett2@usnews.com
4	Sinclare	sdomeney3@irs.gov
5	Park	pmacquire4@sogou.com
6	Pauly	polsen5@techcrunch.com
7	Pavlov	pbygraves6@youtube.com
8	Domingo	dmelland7@is.gd
9	Reed	rfenning8@prlog.org
10	Sholom	schadburn9@etsy.com

Output

```
db.MOCK_DATA.find({}, { id: 1, first_name: 1, email: 1, gender: 1, ip_address: 1, a_big_number: 1, _
[
  {
    "id": 1,
    "first_name": "Wells",
    "email": "wbraddick0@yellowbook.com",
    "gender": "Male",
    "ip_address": "37.5.189.141",
    "a_big_number": 172
  },
  {
    "id": 2,
    "first_name": "Phaidra",
    "email": "pdumbellow1@twitter.com",
    "gender": "Female",
    "ip_address": "18.254.21.176",
    "a_big_number": 567
  },
  {
    "id": 3,
    "first_name": "Taffy",
    "email": "tsherrett2@usnews.com",
    "gender": "Female",
    "ip_address": "130.106.253.132",
    "a_big_number": 545
  },
  ...
]
```

Stored Procedures

Naam	Parameters	Doel	Geeft terug
PROC_GET_MATCH_INFO	match_id, season_name, competition_name, start_date, match_day, home_club, out_club	Ophalen van matchinfo.	match_id, season_name, competition_name, start_date, match_day, home_club_name, out_club_name, stadium_name, referee_person_id, ball_possession_home, ball_possession_out, spectators, amount_of_goals, amount_of_corners, amount_of_fouls, amount_of_passes, amount_of_yellow_cards, amount_of_red_cards, amount_of_shots, amount_of_substitutions
PROC_GET_SCORE_EDITION	season_name, competition_name	Ophalen van score per editie.	rank, competition_name, season_name, club_name, GS (gespeeld), W (gewonnen), G (gelijk gespeeld), L (verloren), DV (doelpunten voor), DT (doelpunten tegen), DS (doelpunten saldo), aantal punten
PROC_GET_SELECTION CLUB OF EDITION	club_name, competition_name, season_name	Ophalen selectie clubs van editie.	person_id, first_name, middle_name, last_name, birth_date
PROC_ADD_MATCH_DATA RED CARD	person_id	Toevoegen van rode kaarten aan een match.	N.V.T.
PROC_ADD_MATCH_DATA YELLOW CARD	person_id	Toevoegen van gele kaarten aan een match.	N.V.T.
PROC_ADD_MATCH_DATA PASS	person_id, success	Toevoegen van passes aan een match.	N.V.T.
PROC_ADD_MATCH_DATA GOAL	person_id	Toevoegen van goals aan een match.	N.V.T.
PROC_ADD_MATCH_DATA SHOT	person_id, on_goal	Toevoegen van shots aan een match.	N.V.T.
PROC_ADD_MATCH_DATA FOUL	person_id	Toevoegen van overtredingen aan een match.	N.V.T.
PROC_ADD_MATCH_DATA CORNER	person_id	Toevoegen van corners aan een match.	N.V.T.
PROC_ADD_MATCH_DATA SUBSTITUTE	in_person_id, out_person_id	Toevoegen van wissels aan een match.	N.V.T.
PROC_ADD_MATCHES TO MATCHDAY	season_name, date, games, club_pairs_table, start	Toevoegen van match aan een matchday.	N.V.T.
PROC_GENERATE_ROUND ROBIN TOURNAMENT TABLE		Aanroepen van tournament door middel van een tabel met daarin club namen. Deze tabel wordt meegegeven als parameter.	

Naam	Parameters	Doel	Geeft terug
PROC_START_NEW_EDITION	competition_name, season_name, club_names_table, date, length, games, amount_of_matchdays_per_round	Starten van een nieuwe editie.	N.V.T.
PROC_INSERT_NEW_COACH	first_name, last_name, middle_name, birth_date	Toevoegen van een nieuwe coach.	N.V.T.
PROC_INSERT_NEW_REFeree	first_name, last_name, middle_name, birth_date	Toevoegen van een nieuwe scheidsrechter.	N.V.T.
PROC_INSERT_NEW_PLAYER	first_name, last_name, middle_name, birth_date, club_name, jersey_number	Toevoegen van een nieuwe speler.	N.V.T.
PROC_START_NEW_EDITION_KO	competition_name, season_name, list_of_clubs, start_date_competition	Starten van een nieuwe editie voor KO.	N.V.T.

Koppeling stored procedures met use cases

Use case	Stored procedure
Ophalen matchinfo	PROC_GET_MATCH_INFO
Ophalen top-lijst	PROC_GET_SCORE_EDITION
Ophalen clubinfo	PROC_GET_SELECTION_CLUB_OF_EDITION
Invoeren matchdata	PROC_ADD_MATCHDATA_RED_CARD
Invoeren matchdata	PROC_ADD_MATCHDATA_YELLOW_CARD
Invoeren matchdata	PROC_ADD_MATCHDATA_PASS
Invoeren matchdata	PROC_ADD_MATCHDATA_GOAL
Invoeren matchdata	PROC_ADD_MATCHDATA_SHOT
Invoeren matchdata	PROC_ADD_MATCHDATA_FOUL
Invoeren matchdata	PROC_ADD_MATCHDATA_CORNER
Invoeren matchdata	PROC_ADD_MATCHDATA_SUBSTITUTE
Invoeren matchdata	PROC_ADD_MATCHES_TO_MATCHDAY
Start nieuw seizoen competitie	PROC_GENERATE_ROUND_ROBIN_TOURNAMENT_TABLE
Start nieuw seizoen competitie	PROC_START_NEW_EDITION
Start nieuw seizoen KO	PROC_START_NEW_EDITION_KO
Toevoegen nieuw persoon	PROC_INSERT_NEW_COACH
Toevoegen nieuw persoon	PROC_INSERT_NEW_REFeree
Toevoegen nieuw persoon	PROC_INSERT_NEW_PLAYER

Koppeling stored procedures met views

Use case	View
Ophalen clubinfo	VW_SELECT_ALL_PLAYERS_PLAYED_IN_MATCH
Ophalen top-lijst	VW_ALL_PLAYERS_EDITIONS
Ophalen clubinfo	VW_CLUB_INFORMATION
Ophalen speelrondeinfo	VW_GET_PLAYROUND_DATA
Ophalen tussenstand competitie	VW_GET_INTERIM_SCORE_EDITION
Ophalen clubinfo	VW_SHOW_CLUB_INFO
Ophalen brackets KO	VW_VIEW_BRACKETS_KO