General task:

The report on item №1 should include:

• list of entities with a description of their purpose;

• graphic file of the developed entity-relationship model;

• name of the notation.

The report on item №2 should include:

• a description of the transformation process (for example, “entity A was converted to table A, and the relationship R (M: N) led to the appearance of an additional table R1, etc.);

• graphical database layout with table names (!) And relationships between them.

The report on item №3 should include:

• explanation (justification!) On the compliance of the database scheme with the normal forms NF1, NF2 and NF3. The explanation is to provide functional dependencies that demonstrate the conclusions. In case of discrepancies, provide a description of the necessary changes in the scheme;

• In case of changes in the database schema, provide an updated version of the schema, otherwise - do not specify the schema.

The report on item №4 should include:

• Provide copies of the pgAdmin4 screen that display column names, types, and constraints (available on the Columns and Constraints tabs of the Properties properties of the object tree tables in pgAdmin4).

• provide copies of the pgAdmin4 screen that display the contents of PostgreSQL database tables. The tables in the image must have a name!

Title of the theme:

Team sports competition in football.

Task №1:

List of entities with a description of their purpose:

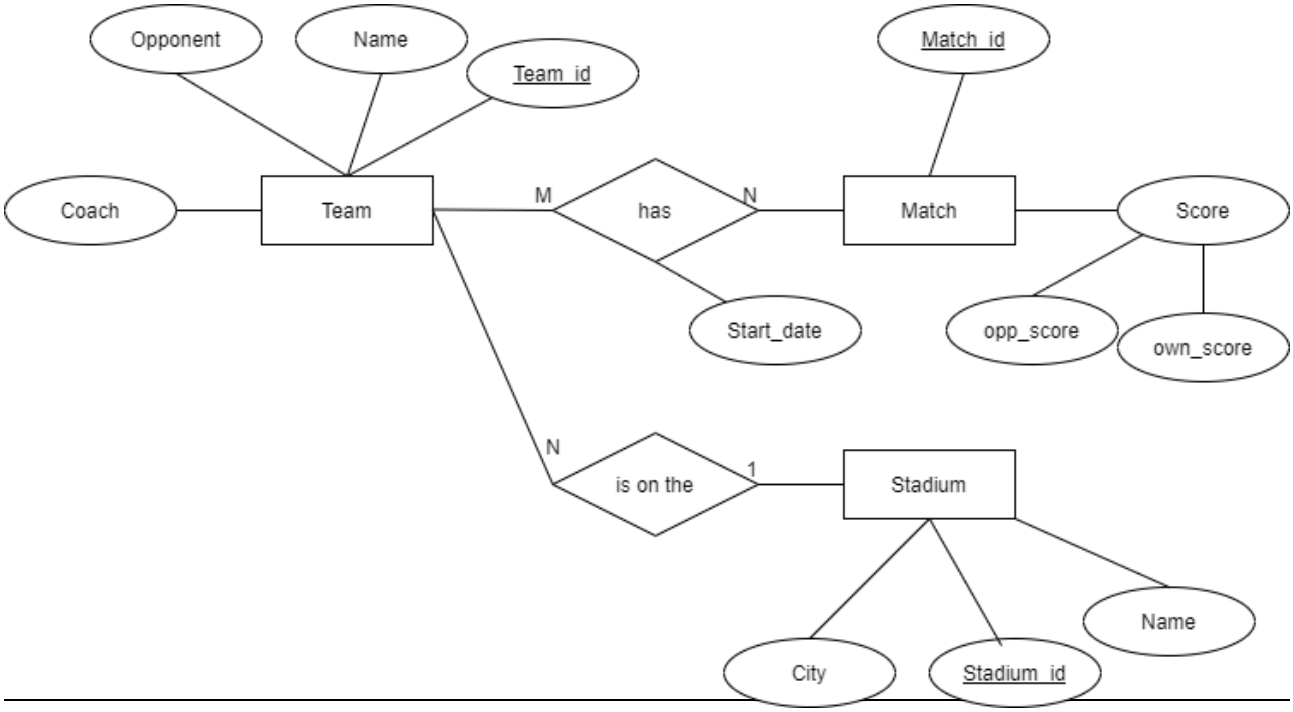
In this work we have 3 entities: Team, Match and Stadium.

The first essence of "Team" is designed to keep track of teams through their unique identification. Also to determine the name of the team and their coach and find out the name of the opponent's team.

The second entity of "Match" is designed to keep track of matches by uniquely identifying them and determining their own account and the account of opponents.

The third entity of "Stadium" is designed to keep track of stadiums by uniquely identifying them, determining the name of the stadium and finding out in which city it is located.

Graphic file of the developed entity-relationship model:



Chen notation used

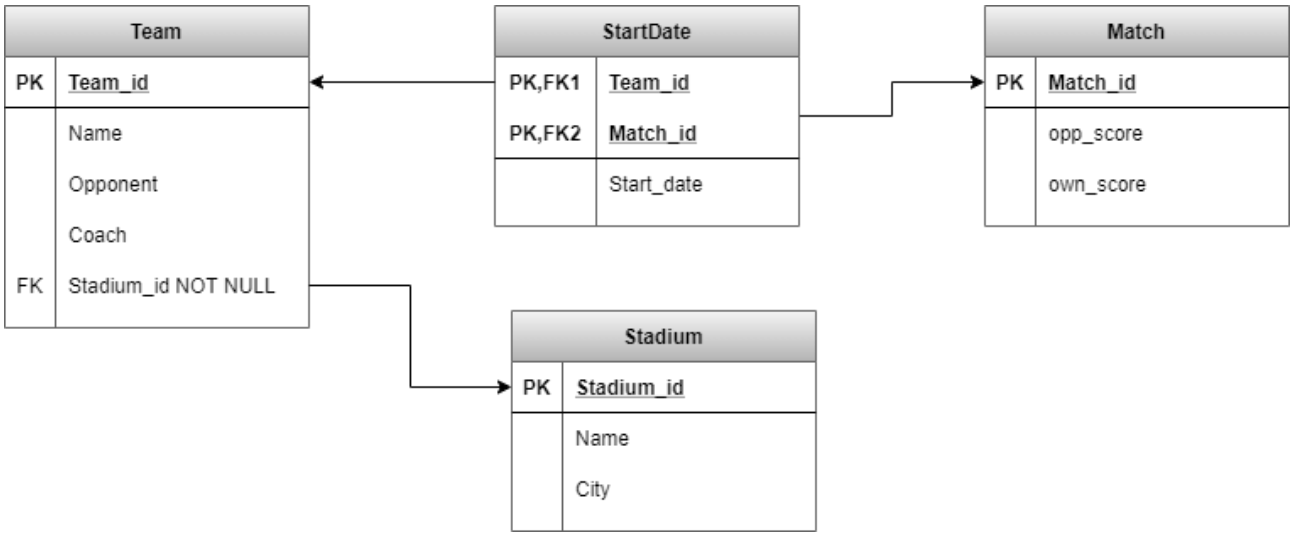
Task №2:

Description of the transformation process:

The “Team” entity was transformed into the “Team” table, the “Match” entity was transformed into the “Match” table, and the “Stadium” entity was transformed into the “Stadium” table. We transfer all attributes that belonged to the entities in the appropriate tables. Instead of the composite entity Score, we write simple atomic attributes opp_score and own_score in the “Match” table. All primary entity keys become primary table keys. Link 1: N "is on the" is required, so the primary entity key on side 1 is added as an attribute to the entity on side N and becomes a

foreign key. And the "has" relationship causes an additional "StartDate" table to appear. It has the keys of both entities and the "Start_date" attribute, which appeared due to the connection with the attribute.

Schematic of the database in a graphical form:



Task №3:

Explanation of the compliance of the database schema with normal forms:

The database schema satisfies the conditions of the first normal form, because all attributes in the table are atomic, there are no ambiguous and composite attributes, each table has a Primary key.

The database schema satisfies the conditions of the second normal form, since all non-key attributes must depend on the complete primary key. To be in the second normal form, all non-key attributes must depend on the integer key. Thus, each relationship that is in the first normal form with one attribute key automatically transitions to the second normal form. In this case, there is a composite key that consists of two attributes, so all non-key attributes must depend on all components of the key. And since Start_date depends on the team that has the match, so we can say that this scheme is in 2NF.

The database schema satisfies the conditions of the third normal form because it satisfies the 2NF conditions and the attributes do not depend on other attributes that

are not primary keys, ie there are no transitive functional dependencies of non-key attributes on key ones.

StartDate(Start_date, Match_id, Team_id)

Primary keys: Match_id, Team_id.

Functional dependencies:

Match_id, Team_id - > Start_date

Team(Team_id, Name, Opponent, Coach, Stadium_id)

Primary keys: Team_id.

Functional dependencies:

Team_id - > Name

Team_id - > Opponent

Team_id - > Coach

Team_id - > Stadium_id

Team_id - > Name, Opponent, Coach, Stadium_id

Coach - > Name

Name - > Coach

Match(Match_id, opp_score, own_score)

Primary keys: Match_id.

Functional dependencies:

Match_id - > opp_score, own_score

Match_id - > opp_score

Match_id - > own_score

Stadium(Stadium_id, Name, City)

Primary keys: Stadium_id

Functional dependencies:

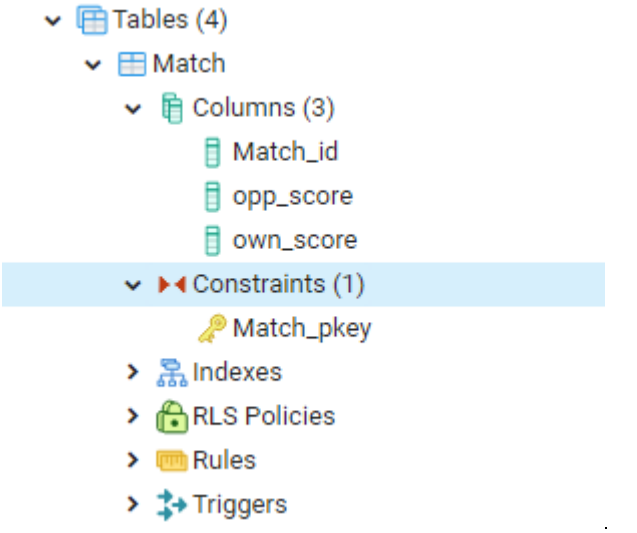
Stadium_id - > Name, City

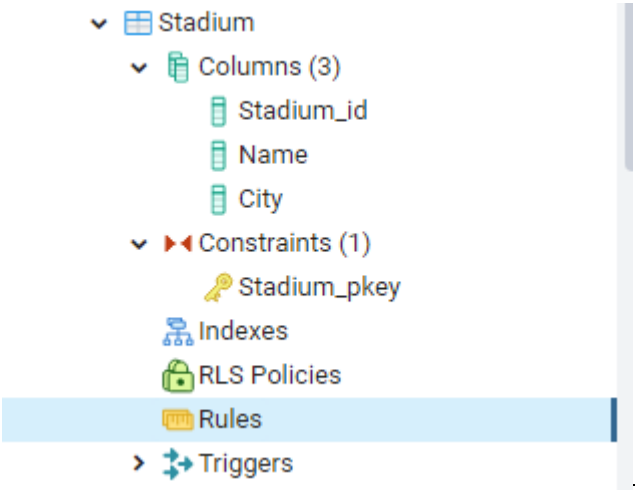
Stadium_id - > Name

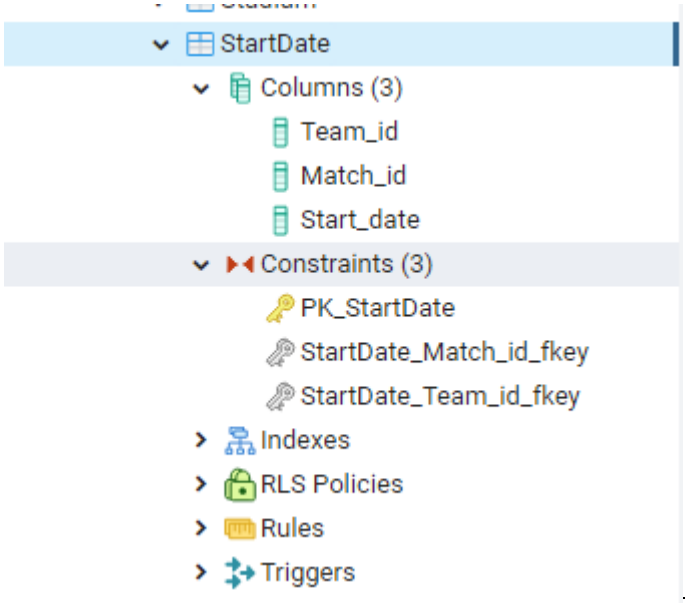
Stadium_id - > City

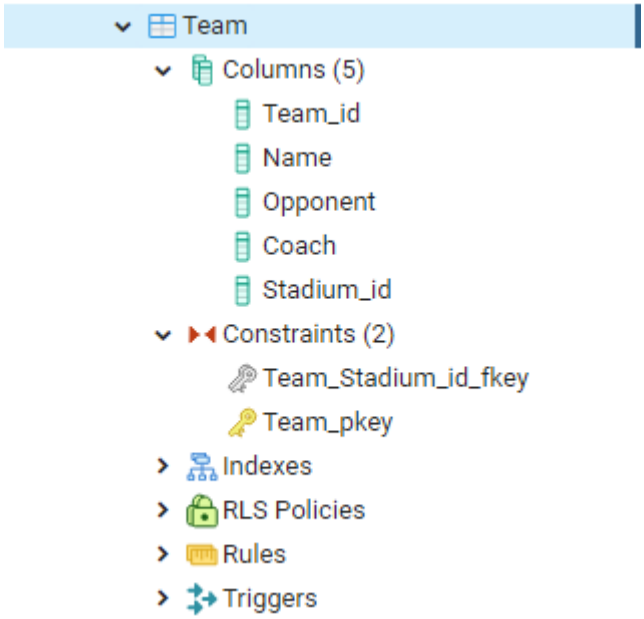
Task №4:

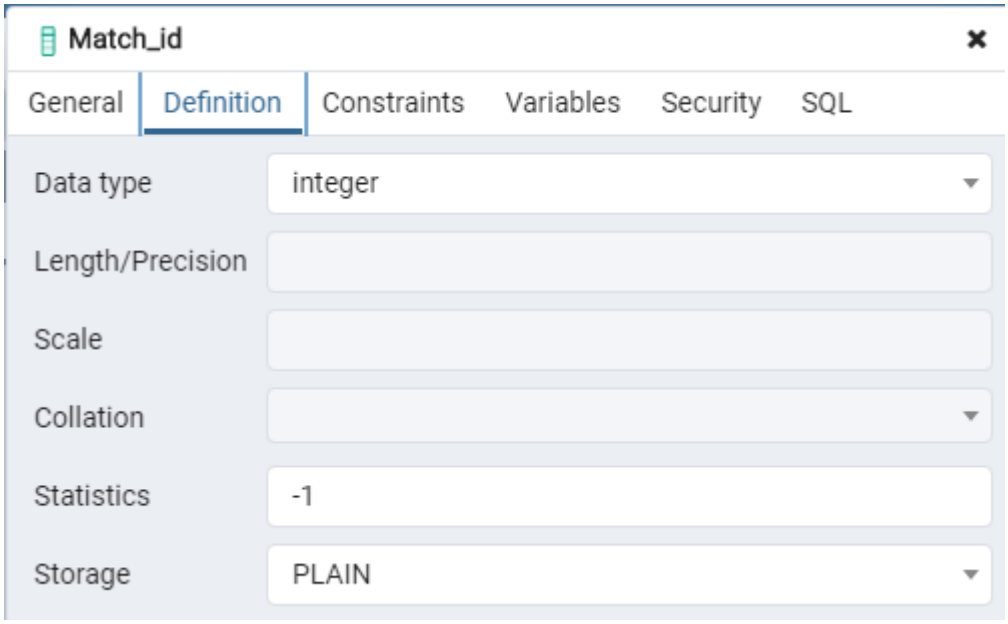
Copies of the pgAdmin4 screen showing column names and types and column restrictions:

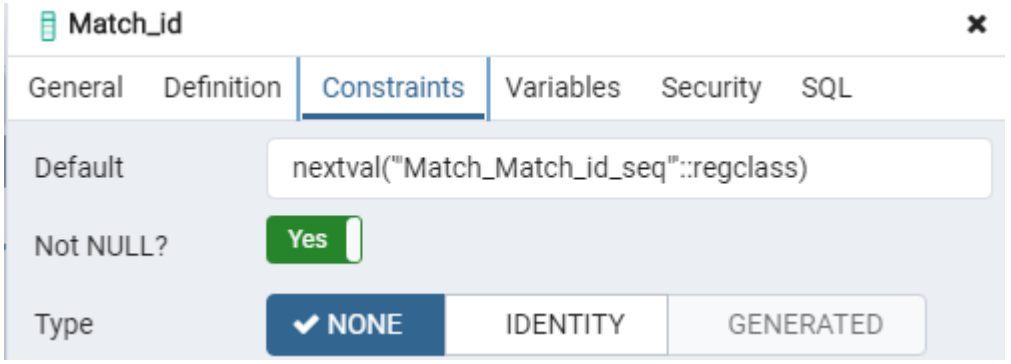


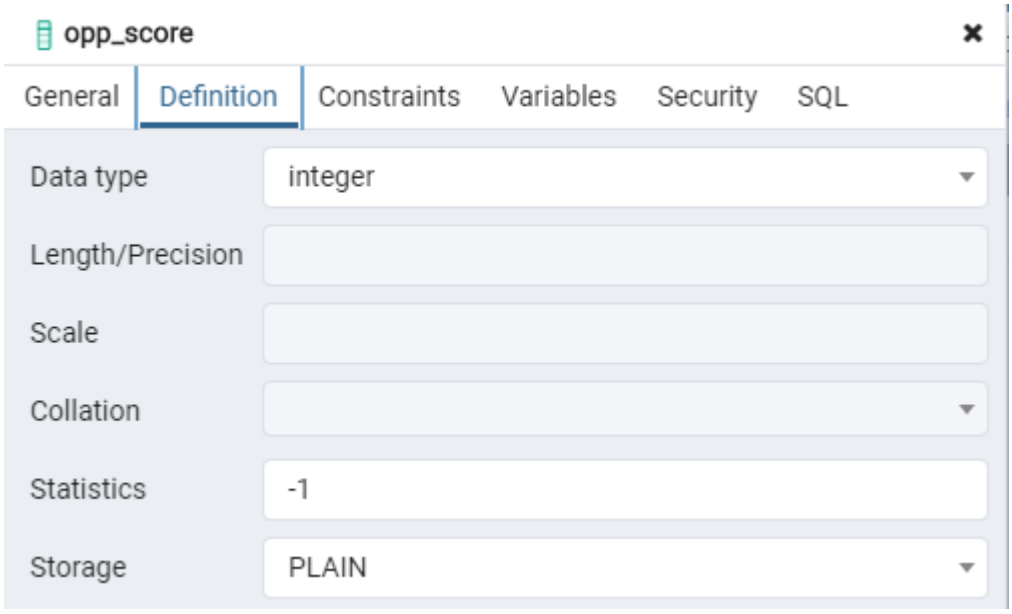


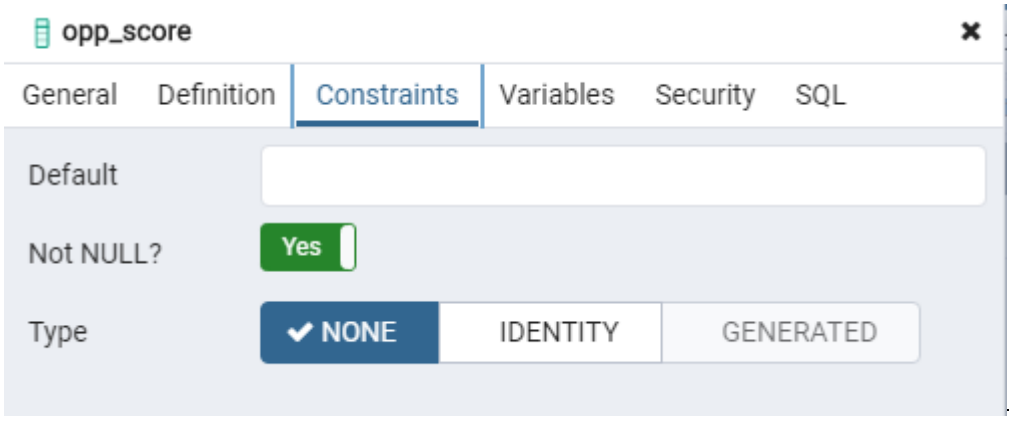

StartDate
Columns (3)
Team_id
Match_id
Start_date
Constraints (3)
PK_StartDate
StartDate_Match_id_fkey
StartDate_Team_id_fkey
Indexes
RLS Policies
Rules
Triggers

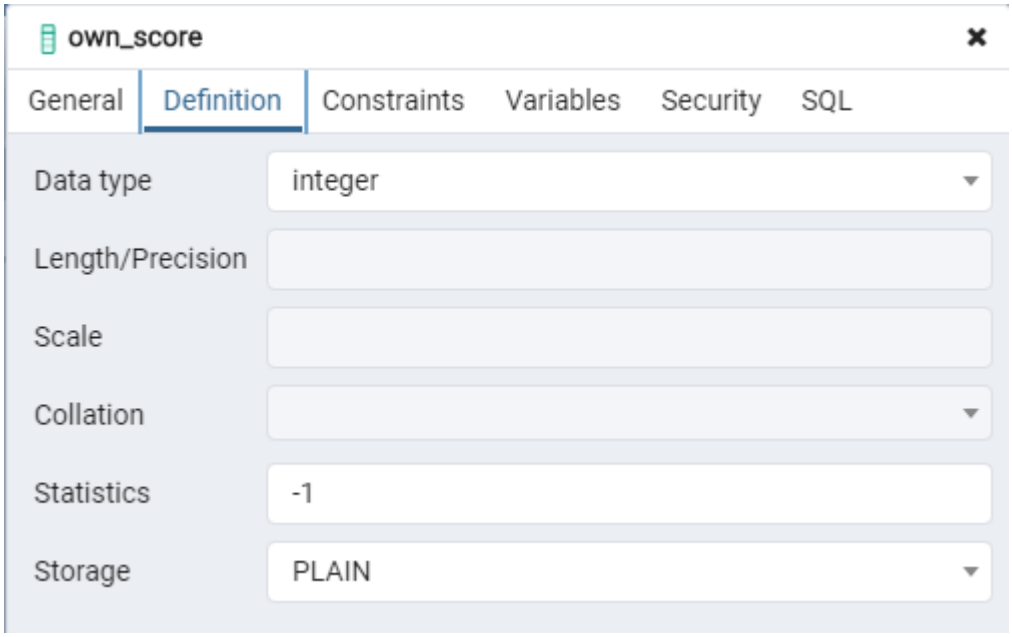

Team
Columns (5)
Team_id
Name
Opponent
Coach
Stadium_id
Constraints (2)
Team_Stadium_id_fkey
Team_pkey
Indexes
RLS Policies
Rules
Triggers

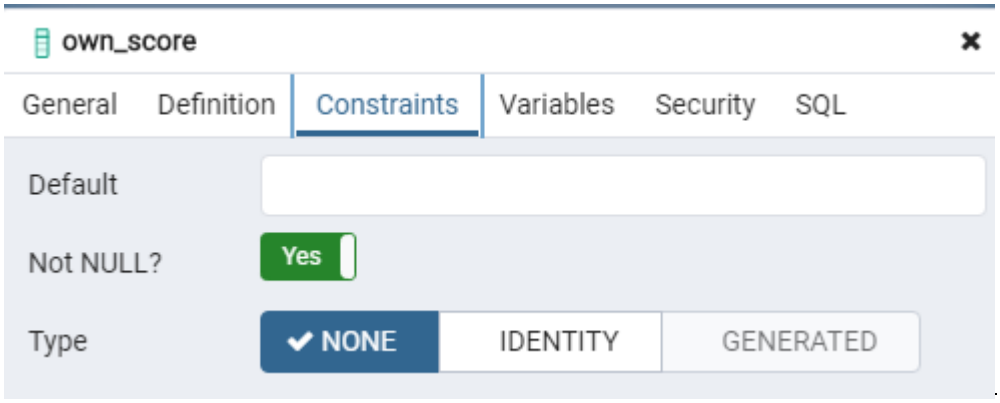

Match_id
General
Definition
Constraints
Variables
Security
SQL
Data type
integer
Length/Precision
Scale
Collation
Statistics
-1
Storage
PLAIN

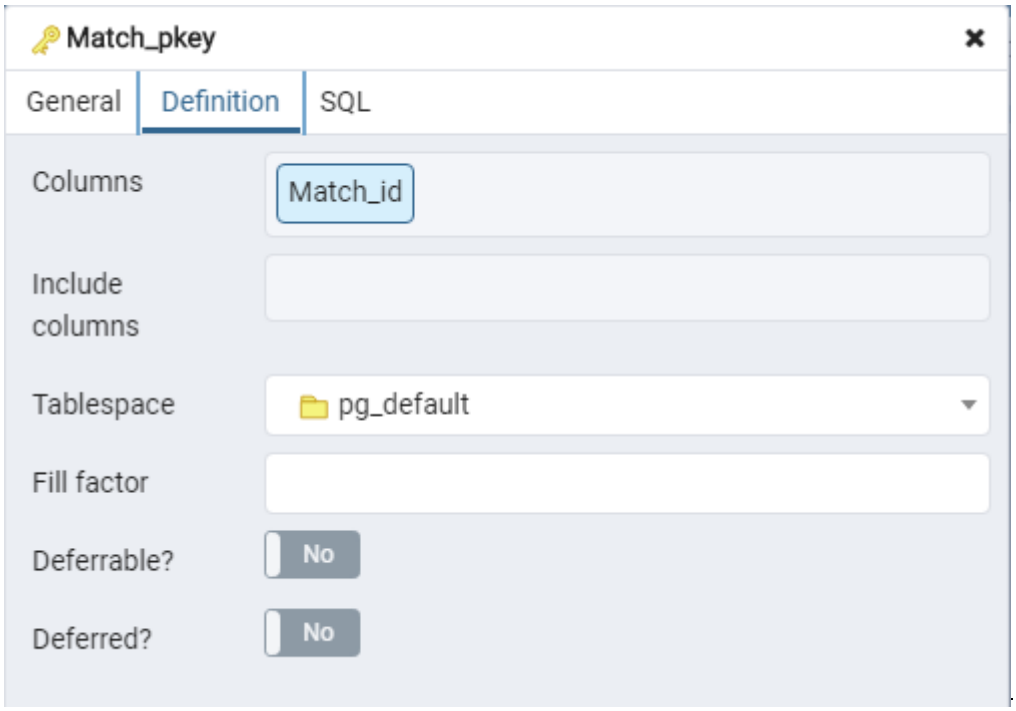

Match_id
General
Definition
Constraints
Variables
Security
SQL
Default
nextval('"Match_Match_id_seq"'::regclass)
Not NULL?
Yes
Type
NONE
IDENTITY
GENERATED

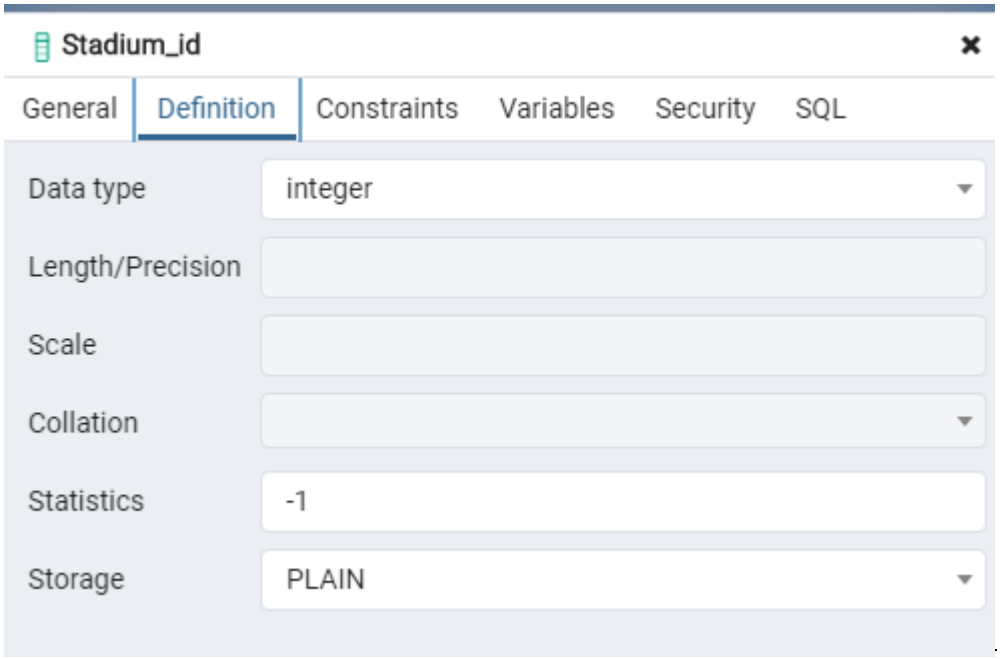

opp_score
General
Definition
Constraints
Variables
Security
SQL
Data type
integer
Length/Precision
Scale
Collation
Statistics
-1
Storage
PLAIN


opp_score
General
Definition
Constraints
Variables
Security
SQL
Default
Not NULL?
Yes
Type
NONE
IDENTITY
GENERATED

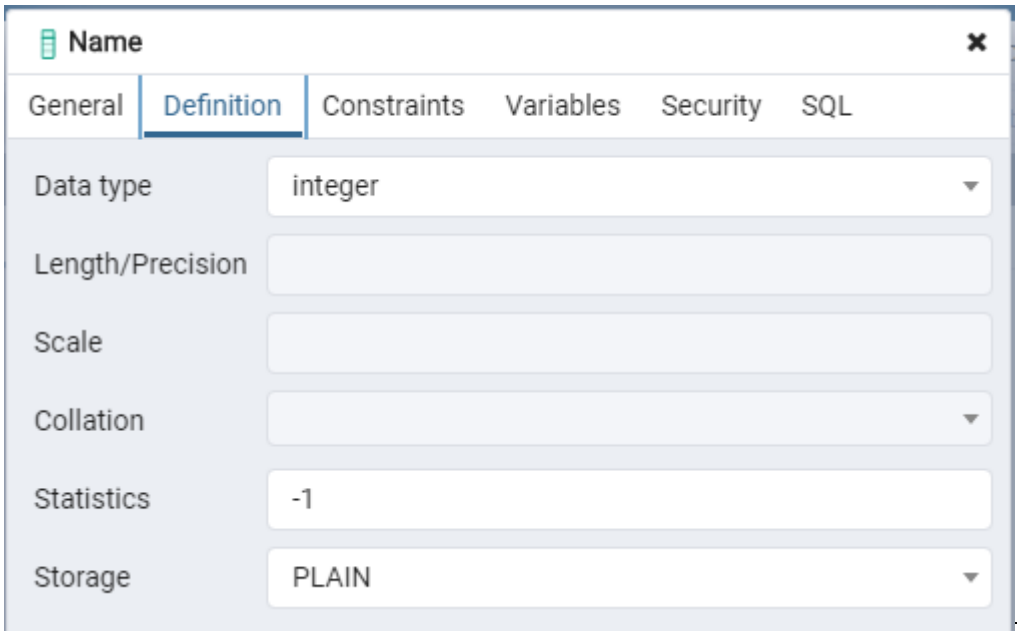

own_score
General
Definition
Constraints
Variables
Security
SQL
Data type
integer
Length/Precision
Scale
Collation
Statistics
-1
Storage
PLAIN

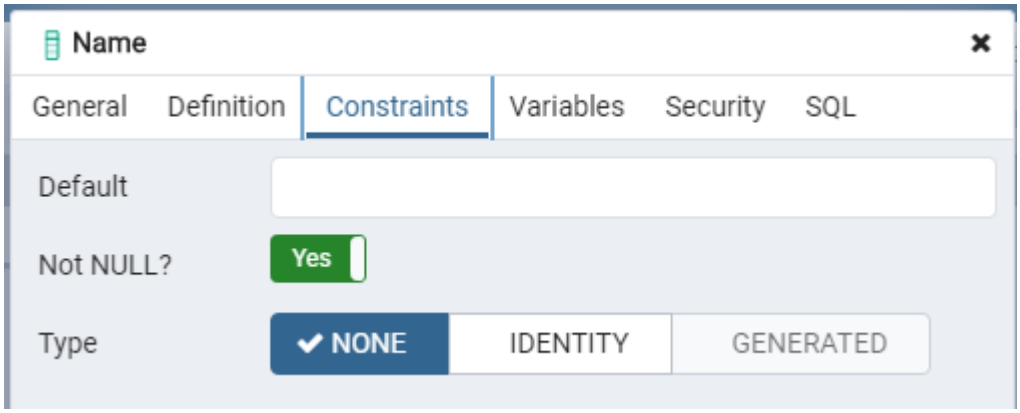

own_score
General
Definition
Constraints
Variables
Security
SQL
Default
Not NULL?
Yes
Type
NONE
IDENTITY
GENERATED

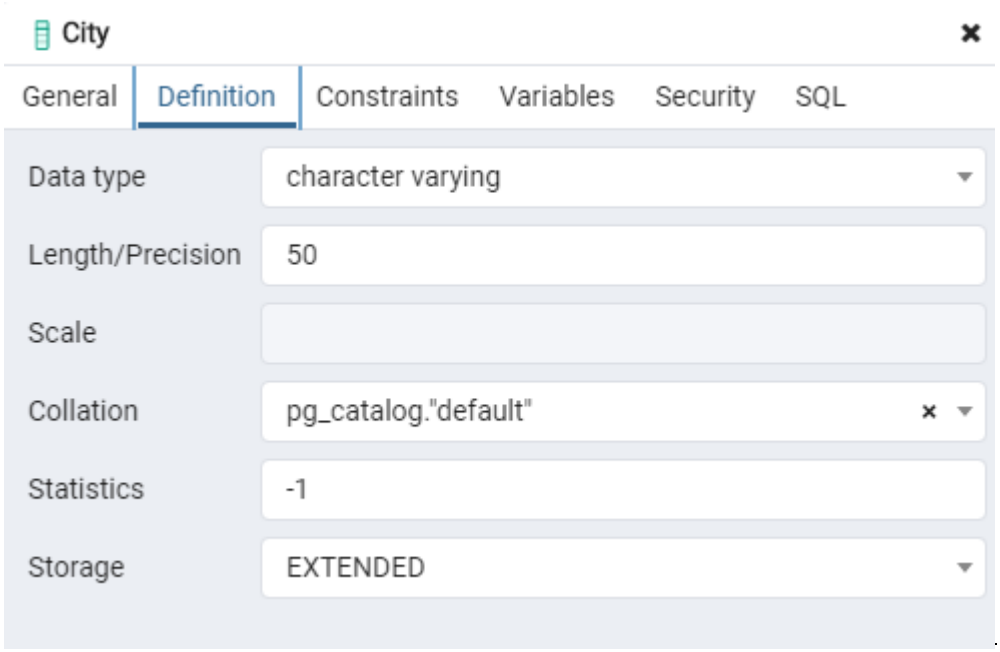

Match_pkey
General
Definition
SQL
Columns
Match_id
Include columns
Tablespace
pg_default
Fill factor
Deferrable?
No
Deferred?
No

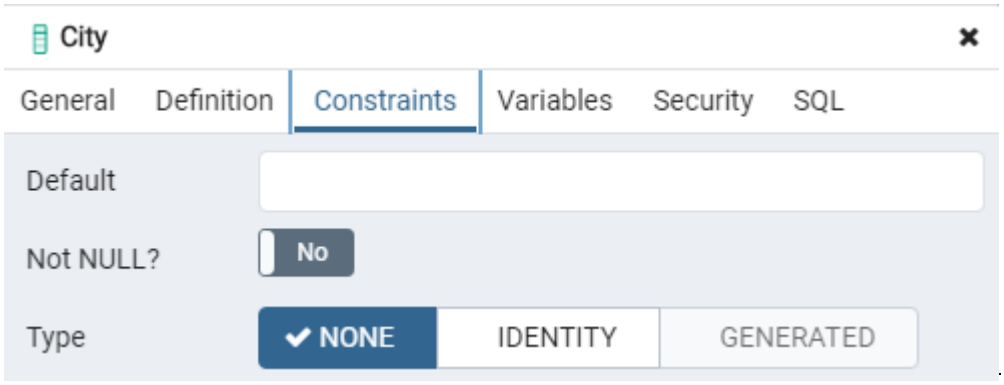

Stadium_id
General
Definition
Constraints
Variables
Security
SQL
Data type
integer
Length/Precision
Scale
Collation
Statistics
-1
Storage
PLAIN

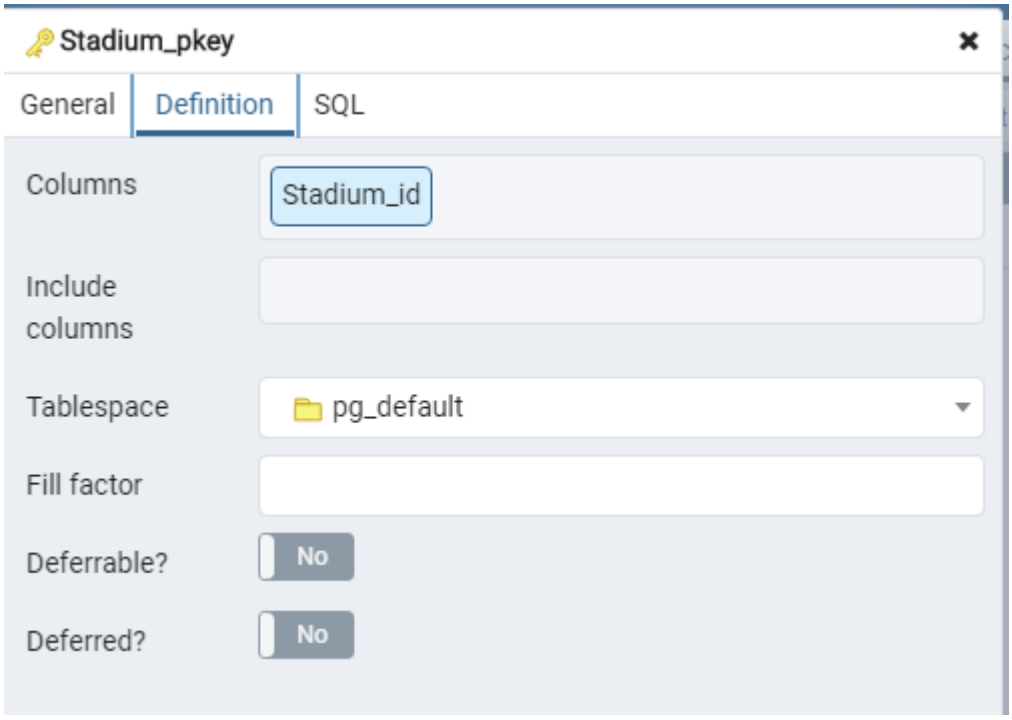

Stadium_id
General
Definition
Constraints
Variables
Security
SQL
Default
nextval('"Match_Match_id_seq"'::regclass)
Not NULL?
Yes
Type
NONE
IDENTITY
GENERATED

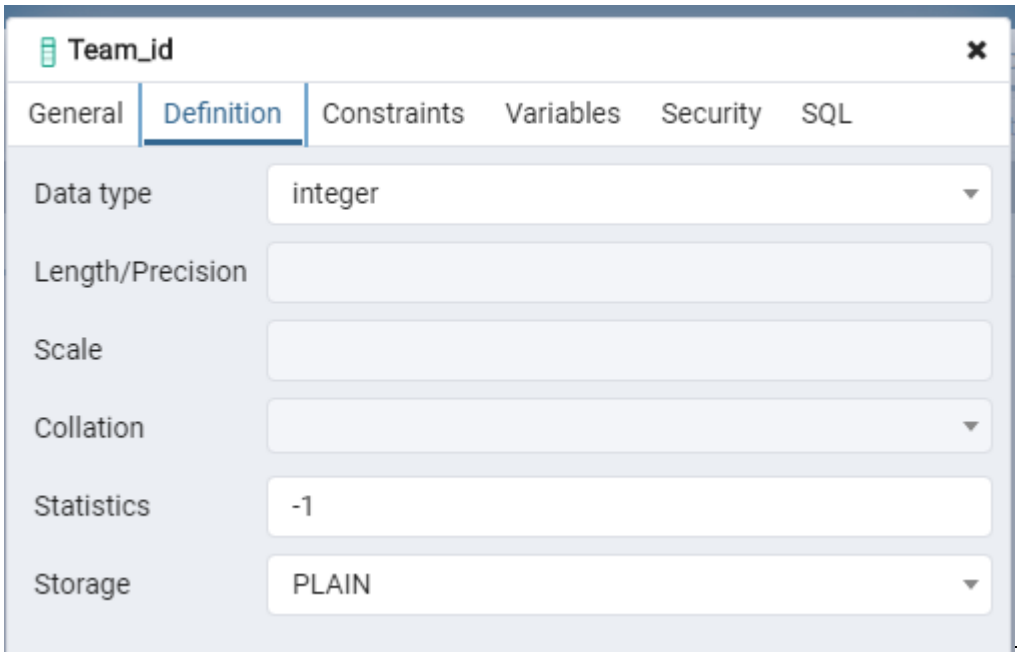

Name
General
Definition
Constraints
Variables
Security
SQL
Data type
integer
Length/Precision
Scale
Collation
Statistics
-1
Storage
PLAIN

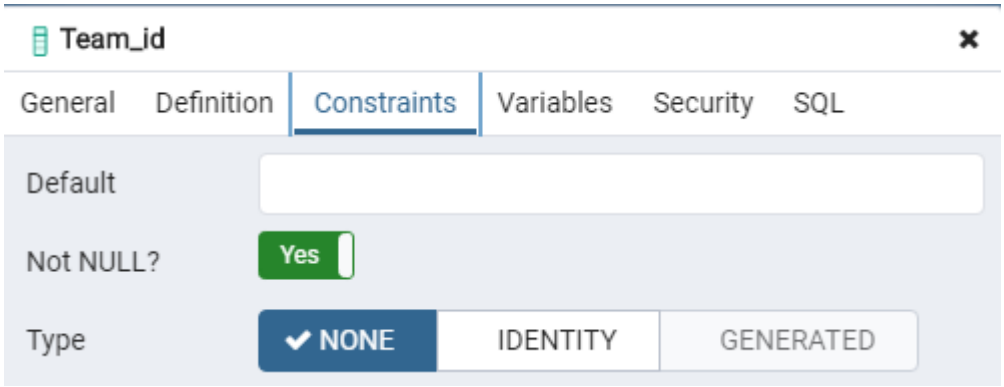

Name
General
Definition
Constraints
Variables
Security
SQL
Default
Not NULL?
Yes
Type
NONE
IDENTITY
GENERATED

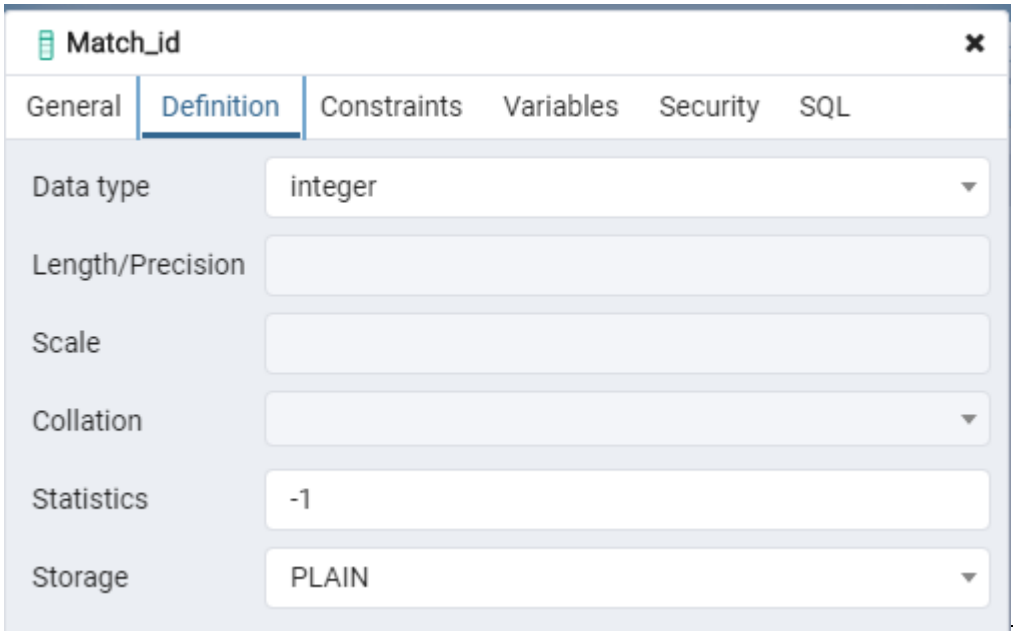

City
General
Definition
Constraints
Variables
Security
SQL
Data type
character varying
Length/Precision
50
Scale
Collation
pg_catalog."default"
Statistics
-1
Storage
EXTENDED

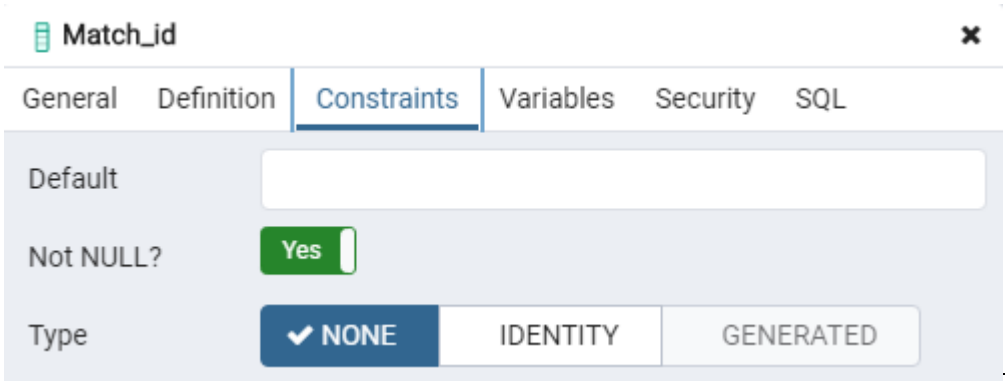

City
General
Definition
Constraints
Variables
Security
SQL
Default
Not NULL?
No
Type
NONE
IDENTITY
GENERATED

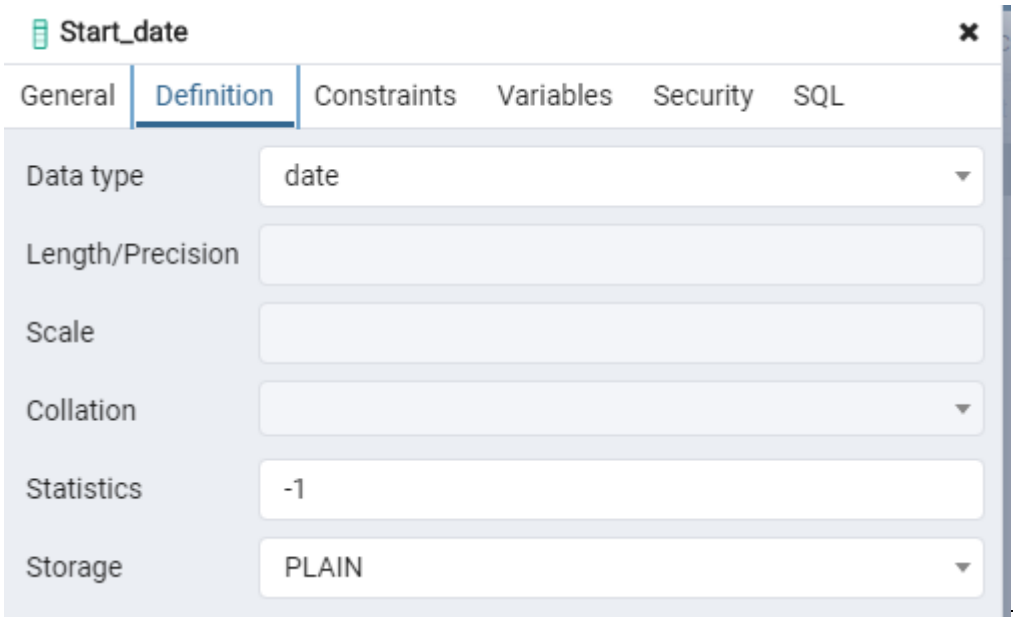

Stadium_pkey
General
Definition
SQL
Columns
Stadium_id
Include columns
Tablespace
pg_default
Fill factor
Deferrable?
No
Deferred?
No

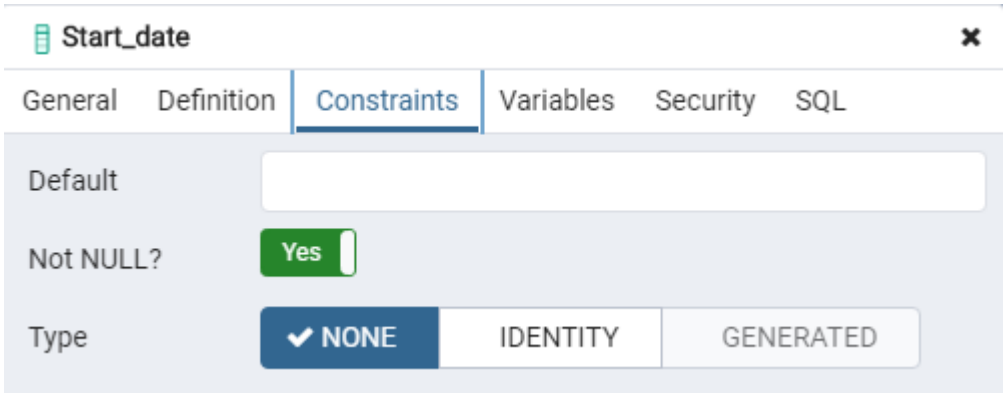

Team_id
General
Definition
Constraints
Variables
Security
SQL
Data type
integer
Length/Precision
Scale
Collation
Statistics
-1
Storage
PLAIN

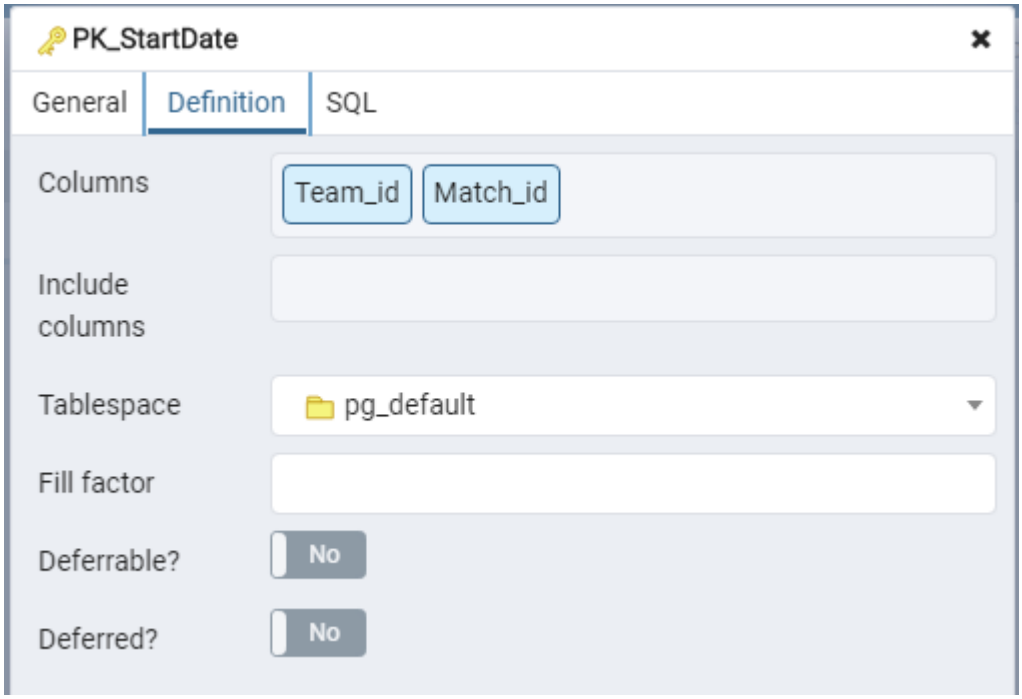

Team_id
General
Definition
Constraints
Variables
Security
SQL
Default
Not NULL?
Yes
Type
NONE
IDENTITY
GENERATED

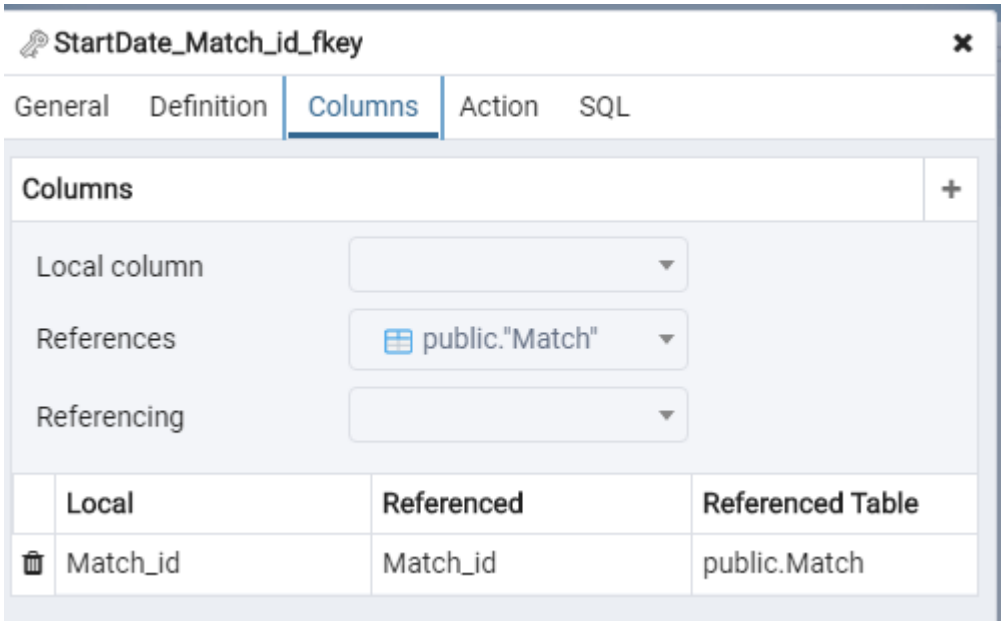

Match_id
General
Definition
Constraints
Variables
Security
SQL
Data type
integer
Length/Precision
Scale
Collation
Statistics
-1
Storage
PLAIN

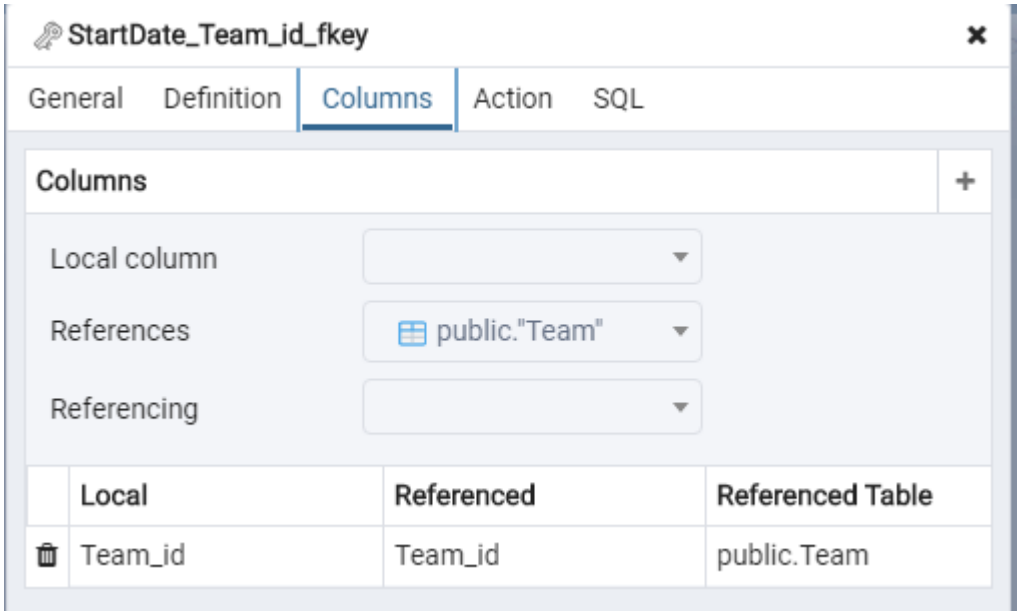

Match_id
General
Definition
Constraints
Variables
Security
SQL
Default
Not NULL?
Yes
Type
NONE
IDENTITY
GENERATED

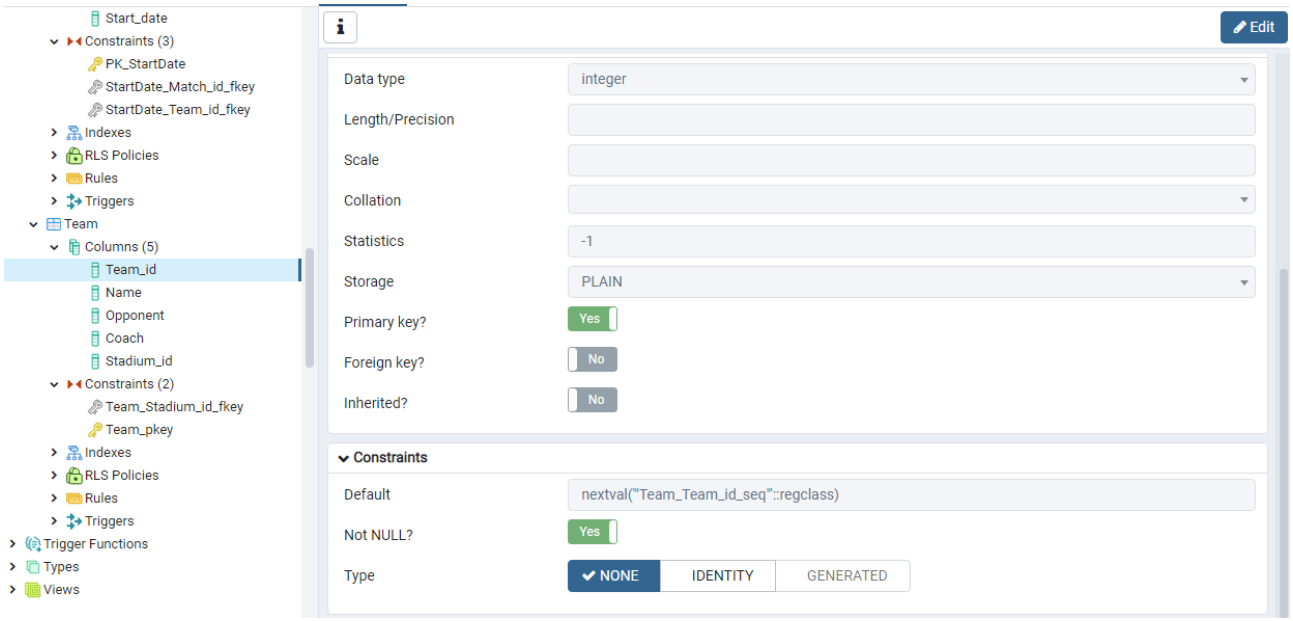

Start_date
General
Definition
Constraints
Variables
Security
SQL
Data type
date
Length/Precision
Scale
Collation
Statistics
-1
Storage
PLAIN

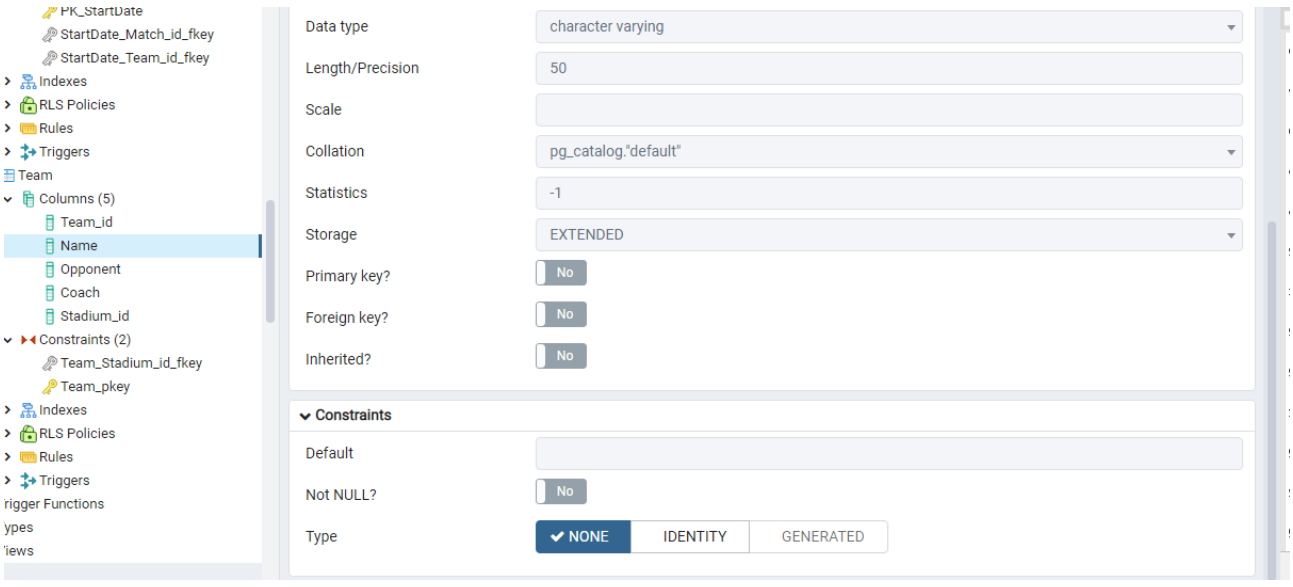

Start_date
General
Definition
Constraints
Variables
Security
SQL
Default
Not NULL?
Yes
Type
NONE
IDENTITY
GENERATED

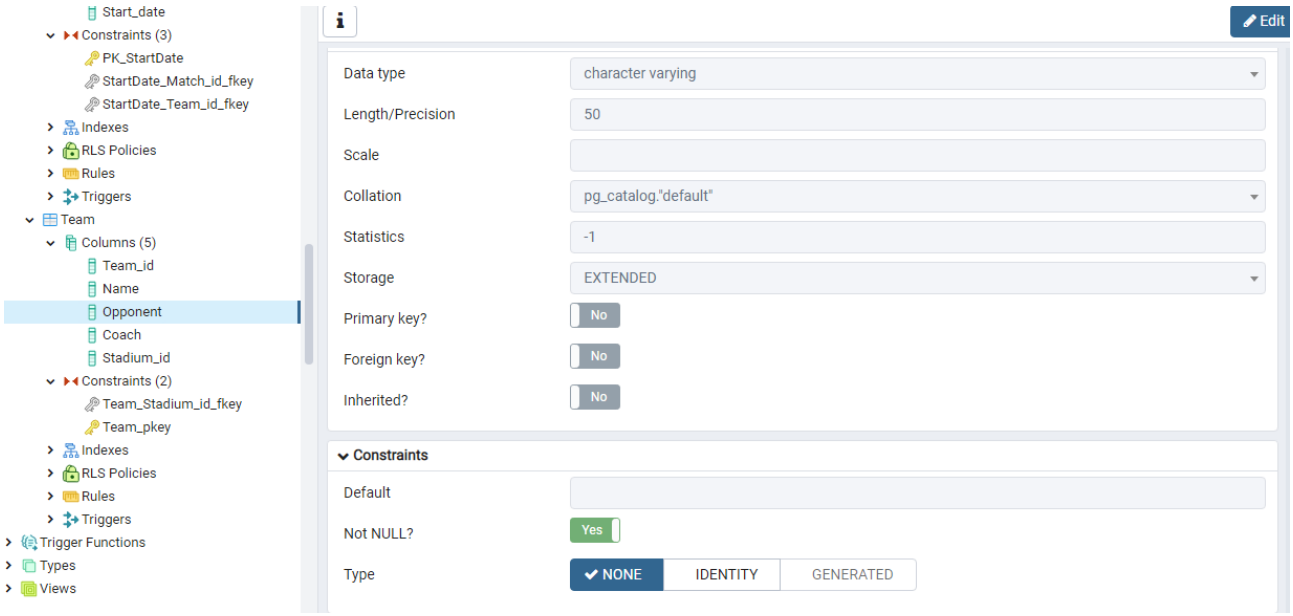

PK_StartDate
General
Definition
SQL
Columns
Team_id
Match_id
Include columns
Tablespace
pg_default
Fill factor
Deferrable?
No
Deferred?
No

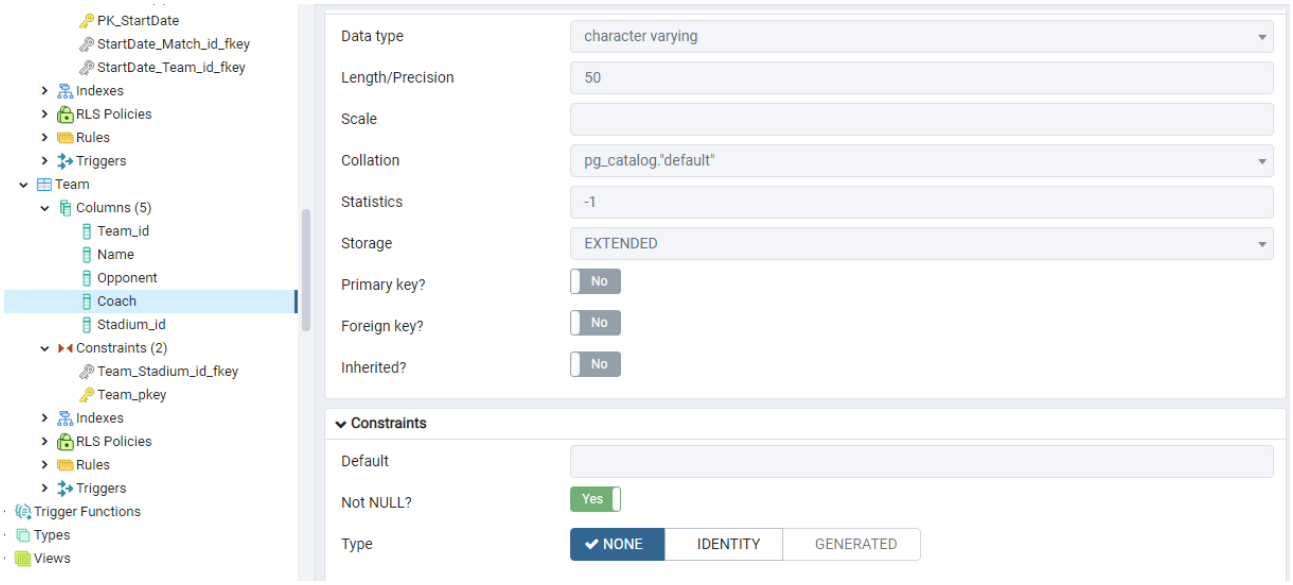

StartDate_Match_id_fkey
General
Definition
Columns
Action
SQL
Columns
Local column
References
public."Match"
Referencing
Local
Referenced
Referenced Table
Match_id
Match_id
public.Match

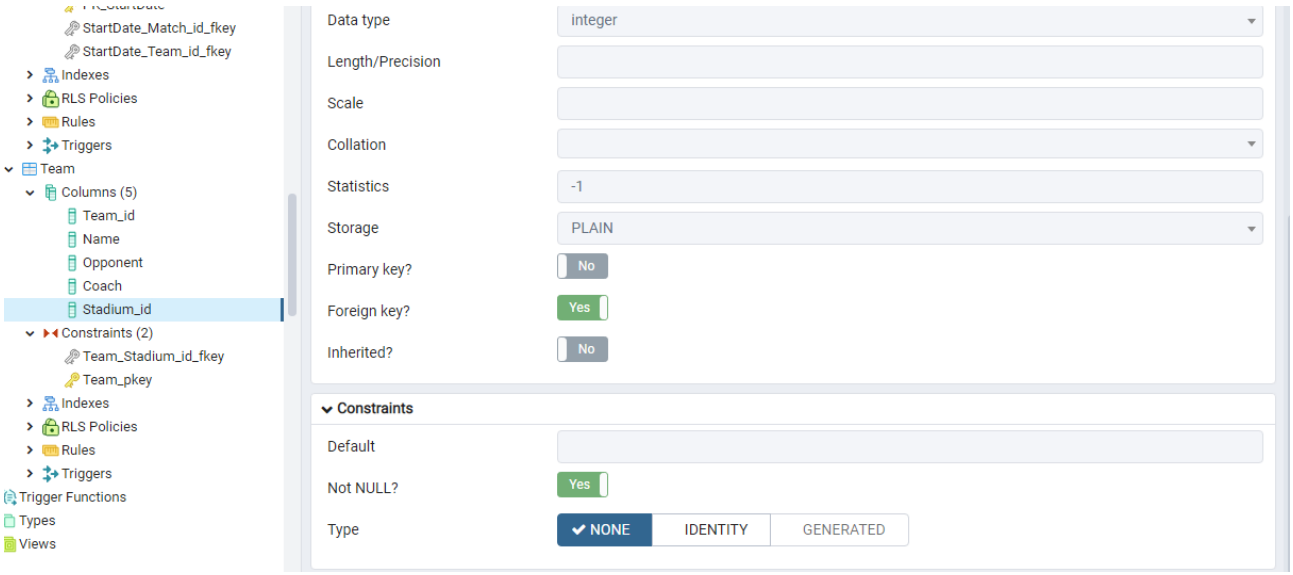

StartDate_Team_id_fkey
General
Definition
Columns
Action
SQL
Columns
Local column
References
public."Team"
Referencing
Local
Referenced
Referenced Table
Team_id
Team_id
public.Team

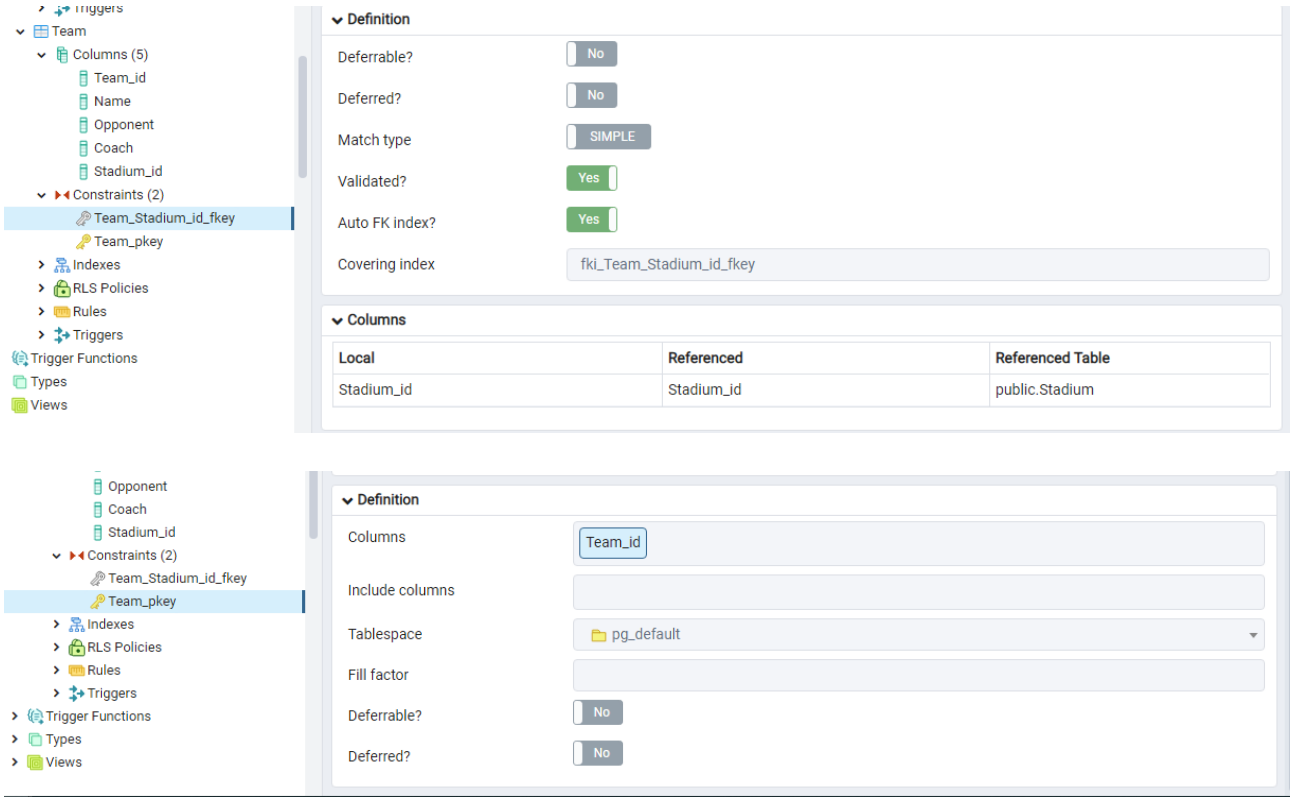

Start_date
Constraints (3)
PK_StartDate
StartDate_Match_id_fkey
StartDate_Team_id_fkey
Indexes
RLS Policies
Rules
Triggers
Team
Columns (5)
Team_id
Name
Opponent
Coach
Stadium_id
Constraints (2)
Team_Stadium_id_fkey
Team_pkey
Indexes
RLS Policies
Rules
Triggers
Trigger Functions
Types
Views
Edit
Data type
integer
Length/Precision
Scale
Collation
Statistics
-1
Storage
PLAIN
Primary key?
Yes
Foreign key?
No
Inherited?
No
Constraints
Default
nextval('"Team_Team_id_seq"'::regclass)
Not NULL?
Yes
Type
NONE
IDENTITY
GENERATED


PK_StartDate
StartDate_Match_id_fkey
StartDate_Team_id_fkey
Indexes
RLS Policies
Rules
Triggers
Team
Columns (5)
Team_id
Name
Opponent
Coach
Stadium_id
Constraints (2)
Team_Stadium_id_fkey
Team_pkey
Indexes
RLS Policies
Rules
Triggers
rigger Functions
ypes
iews
Data type
character varying
Length/Precision
50
Scale
Collation
pg_catalog."default"
Statistics
-1
Storage
EXTENDED
Primary key?
No
Foreign key?
No
Inherited?
No
Constraints
Default
Not NULL?
No
Type
NONE
IDENTITY
GENERATED

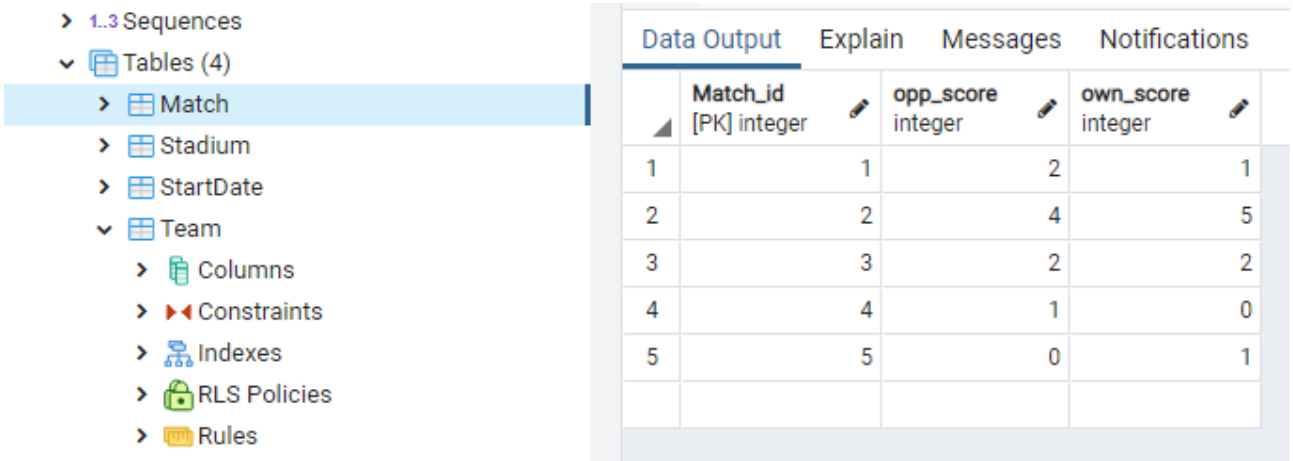

Start_date
Constraints (3)
PK_StartDate
StartDate_Match_id_fkey
StartDate_Team_id_fkey
Indexes
RLS Policies
Rules
Triggers
Team
Columns (5)
Team_id
Name
Opponent
Coach
Stadium_id
Constraints (2)
Team_Stadium_id_fkey
Team_pkey
Indexes
RLS Policies
Rules
Triggers
Trigger Functions
Types
Views
Edit
Data type
character varying
Length/Precision
50
Scale
Collation
pg_catalog."default"
Statistics
-1
Storage
EXTENDED
Primary key?
No
Foreign key?
No
Inherited?
No
Constraints
Default
Not NULL?
Yes
Type
NONE
IDENTITY
GENERATED

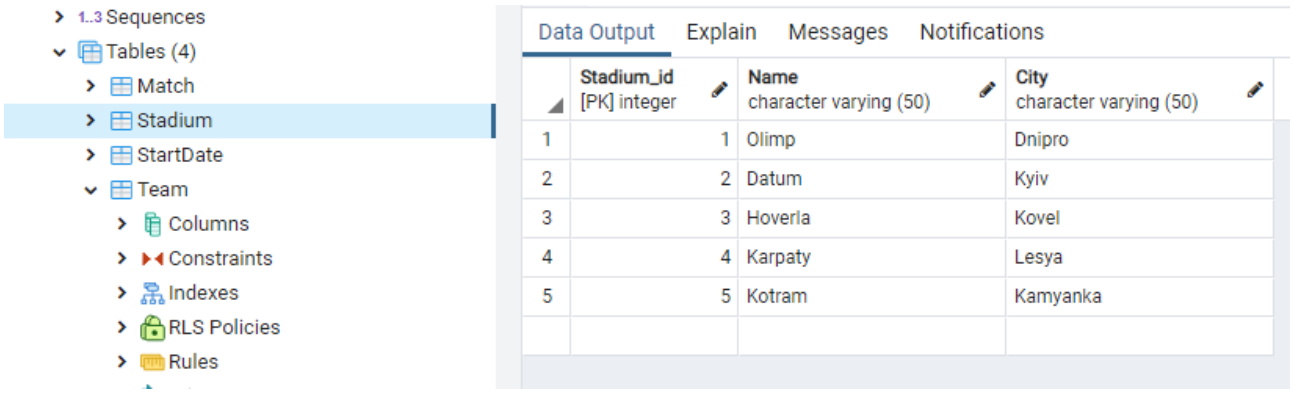

PK_StartDate
StartDate_Match_id_fkey
StartDate_Team_id_fkey
Indexes
RLS Policies
Rules
Triggers
Team
Columns (5)
Team_id
Name
Opponent
Coach
Stadium_id
Constraints (2)
Team_Stadium_id_fkey
Team_pkey
Indexes
RLS Policies
Rules
Triggers
Trigger Functions
Types
Views
Data type
character varying
Length/Precision
50
Scale
Collation
pg_catalog."default"
Statistics
-1
Storage
EXTENDED
Primary key?
No
Foreign key?
No
Inherited?
No
Constraints
Default
Not NULL?
Yes
Type
NONE
IDENTITY
GENERATED

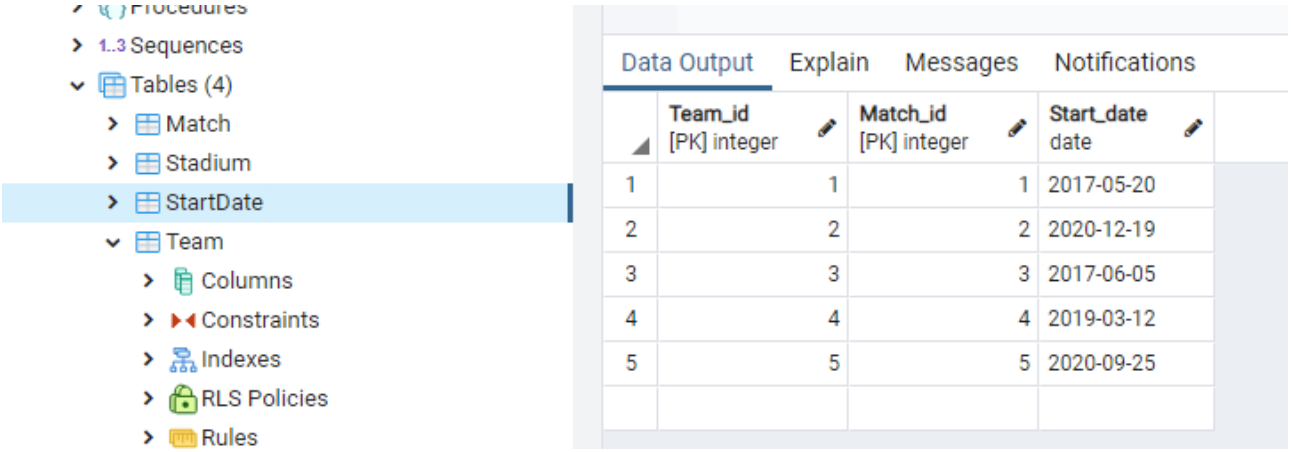

StartDate_Match_id_fkey
StartDate_Team_id_fkey
Indexes
RLS Policies
Rules
Triggers
Team
Columns (5)
Team_id
Name
Opponent
Coach
Stadium_id
Constraints (2)
Team_Stadium_id_fkey
Team_pkey
Indexes
RLS Policies
Rules
Triggers
Trigger Functions
Types
Views
Data type
integer
Length/Precision
Scale
Collation
Statistics
-1
Storage
PLAIN
Primary key?
No
Foreign key?
Yes
Inherited?
No
Constraints
Default
Not NULL?
Yes
Type
NONE
IDENTITY
GENERATED

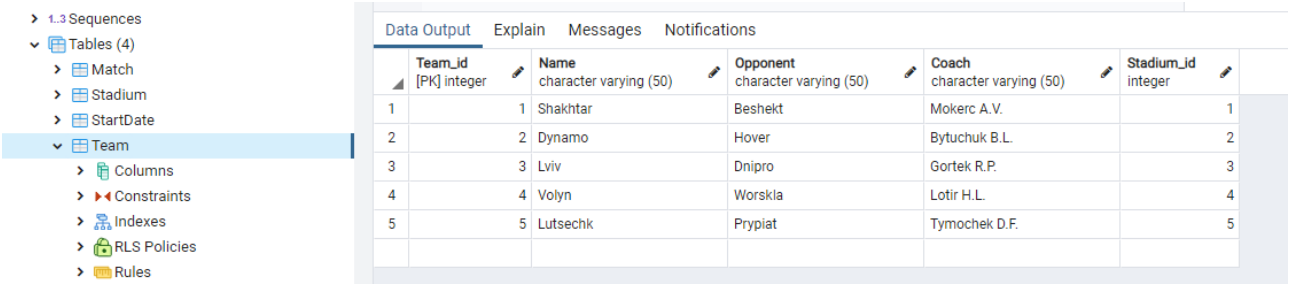


Copies of the pgAdmin4 screen that displays the contents of PostgreSQL database tables:









Frequently asked question:

1. Specify the purpose of entity-relationship charts.

   Entity-relationship diagrams are used to create a database design. This allows you to see the database model at a higher level before its final implementation. This allows you to make sure that everything you need is recorded in the chart and configure it at a logical level before you move on to more complex database creation processes. You can simply show the design to a non-technical client to see if all the requirements have been met. In addition, this type of chart is widely used in the design or analysis of databases in business processes.

2. Name the main objects of the PostgreSQL schema?

   In PostgreSQL, a schema is a namespace that contains named database objects, such as tables, views, indexes, data types, functions, stored procedures, and statements.

3. Give examples of different types of relationships in databases (1: 1, 1: N, N: M).

To each other: one entity is associated with another entity. For example: Each person can have only one passport, or a marriage of a man and a woman. The primary school teacher has only one class.

One to many: one entity is associated with many other entities. For example: the user has many orders, the teacher has many students, the course consists of many groups.

Many to many: Many entities are related to many other entities. For example: a buyer can buy more than one product, and a product can be bought more than once. A teacher can teach to many groups, and a group can learn from many teachers.