



НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ
ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

**Кафедра системного програмування та спеціалізованих комп'ютерних
систем**

Лабораторна робота №1

з дисципліни **Бази даних і засоби управління**

*на тему: “Проектування бази даних та ознайомлення з базовими операціями
СУБД PostgreSQL”*

Виконав:

студент III курсу

групи KB-91

Неживий Б. М.

Перевірив:

Павловський В. І.

Київ – 2021

Метою роботи є здобуття вмінь проектування бази даних та практичних навичок створення реляційних баз даних за допомогою PostgreSQL.

Завдання роботи полягає у наступному:

1. Розробити модель «сутність-зв'язок» предметної галузі, обраної студентом самостійно, відповідно до пункту «Вимоги до ER-моделі».
2. Перетворити розроблену модель у схему бази даних (таблиці) PostgreSQL.
3. Виконати нормалізацію схеми бази даних до третьої нормальної форми (3НФ).
4. Ознайомитись із інструментарієм PostgreSQL та pgAdmin 4 та занести декілька рядків даних у кожну з таблиць засобами pgAdmin 4.

Вимоги до ER-моделі:

1. Сутності моделі предметної галузі мають містити зв'язки типу 1:N або N:M.
2. Кількість сутностей у моделі – 3-4. Кількість атрибутів у кожній сутності: від двох до п'яти.
3. Передбачити наявність зв'язку з атрибутом.

Для побудови ER-діаграм використовувати одну із нотацій: Чена, “Пташиної лапки (Crow’s foot)”, UML.

Опис предметної галузі

При створенні бази даних «Кіберспорт» можна виділити наступні сутності: загальні відомості про ігрову дисципліну (gaming discipline), організацію (organization), команду (team) та гравця (players).

Атрибути наведених сутностей:

- 1) Gaming discipline: title, genre, release date;
- 2) organization: title, region, main sponsor;
- 3) team: title, rating, tier;
- 4) players: name, rating, country.

Опис зв'язків

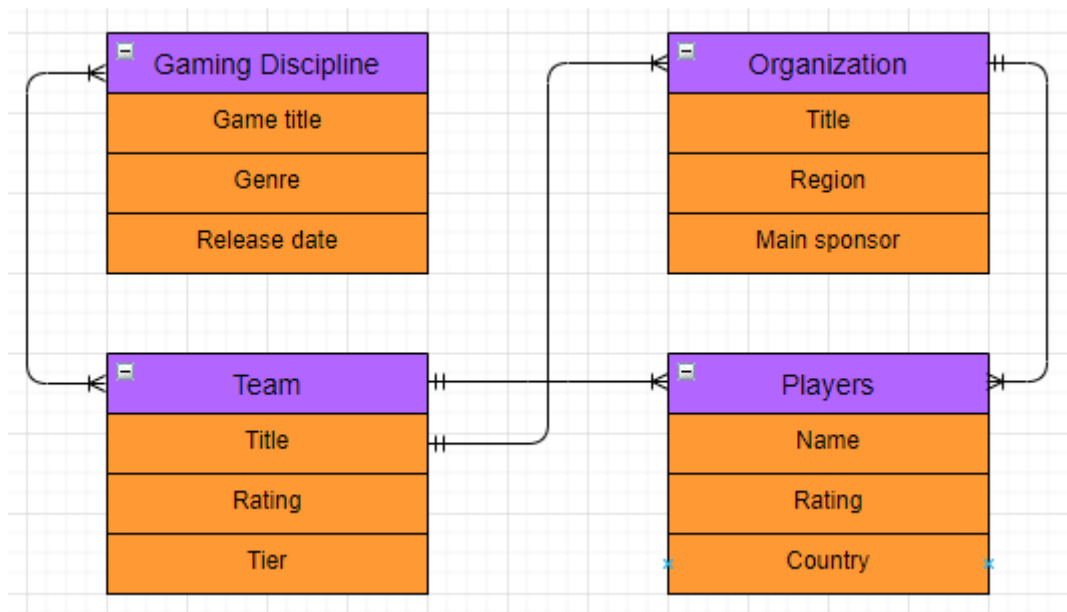
В конкретній ігровій дисципліні грає багато команд, а команди, маючи однакову назву, можуть грати в багато різних дисциплін. Тому відношенням discipline до team буде $N : M$.

В певній організації є багато команд (наприклад основний та молодіжний склади), в той час як команда може виступати тільки від імені однієї організації. Тож зв'язок між organization та team $1 : N$.

Також організація включає в себе велику кількість гравців, що числяться як її представники. А гравці, в свою чергу, можуть бути в складі тільки конкретної організації. Через це зв'язок organization та players $1 : N$.

В одній команді завжди більше 1 гравця, а гравці не можуть одночасно бути членами декількох команд. Тому зв'язок між сутностями team та players $1 : N$.

Концептуальна модель «сутність-зв'язок» предметної галузі «Кіберспорт» зображена на малюнку 1.



Мал. 1 - ER-діаграма побудована за нотацією “Пташиної лапки (Crow’s foot)”. Задана ER-діаграма була побудована у додатку draw.io.

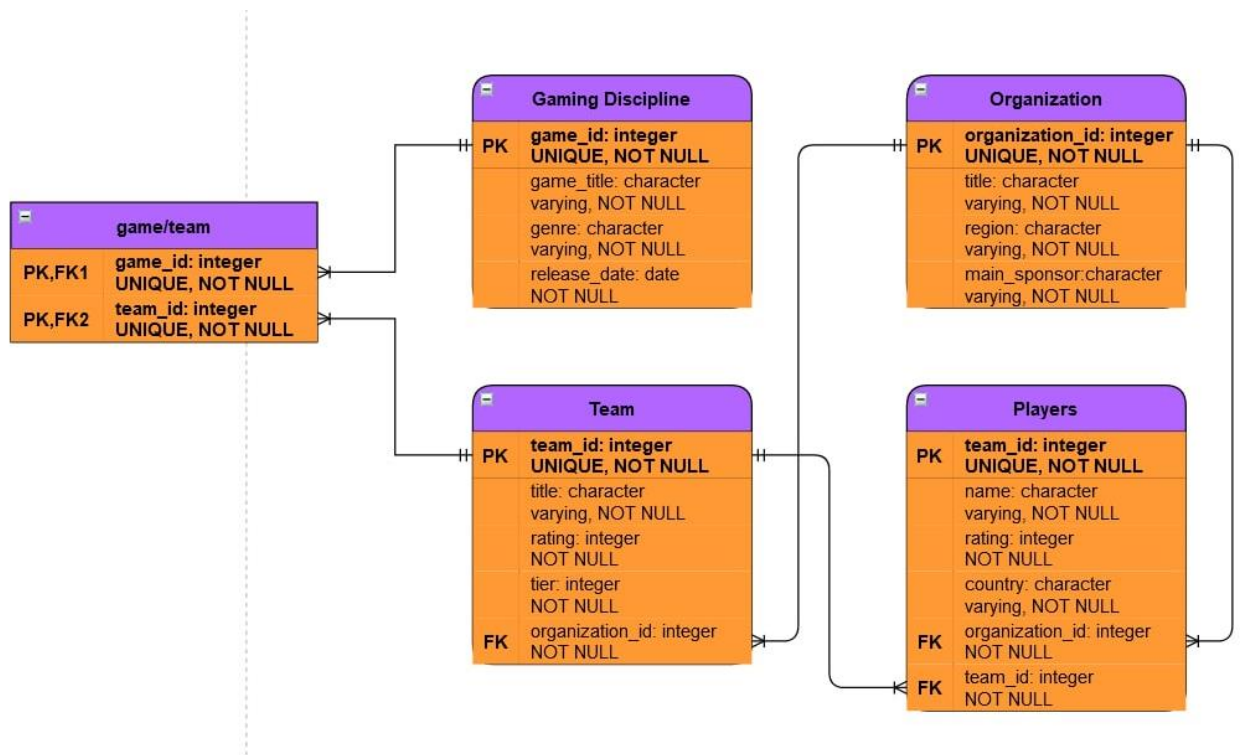
Перетворення концептуальної моделі у схему баз даних

Для кожної сутності створюється таблиця. Причому кожному атрибуту сутності відповідає стовпець таблиці. В даній моделі перетворення в схему баз даних відбувалося за такими правилами:

1. Якщо зв'язок типу 1 : N і клас приналежності сутності на стороні N є обов'язковим, то необхідно побудувати таблицю для кожної сутності. Первинний ключ сутності повинен бути первинним ключем відповідної таблиці. Первинний ключ сутності на стороні 1 додається як атрибут в таблицю для сутності на стороні N. Даний зв'язок можна спостерігати між сутностями organization та team, organization та players, team та players.

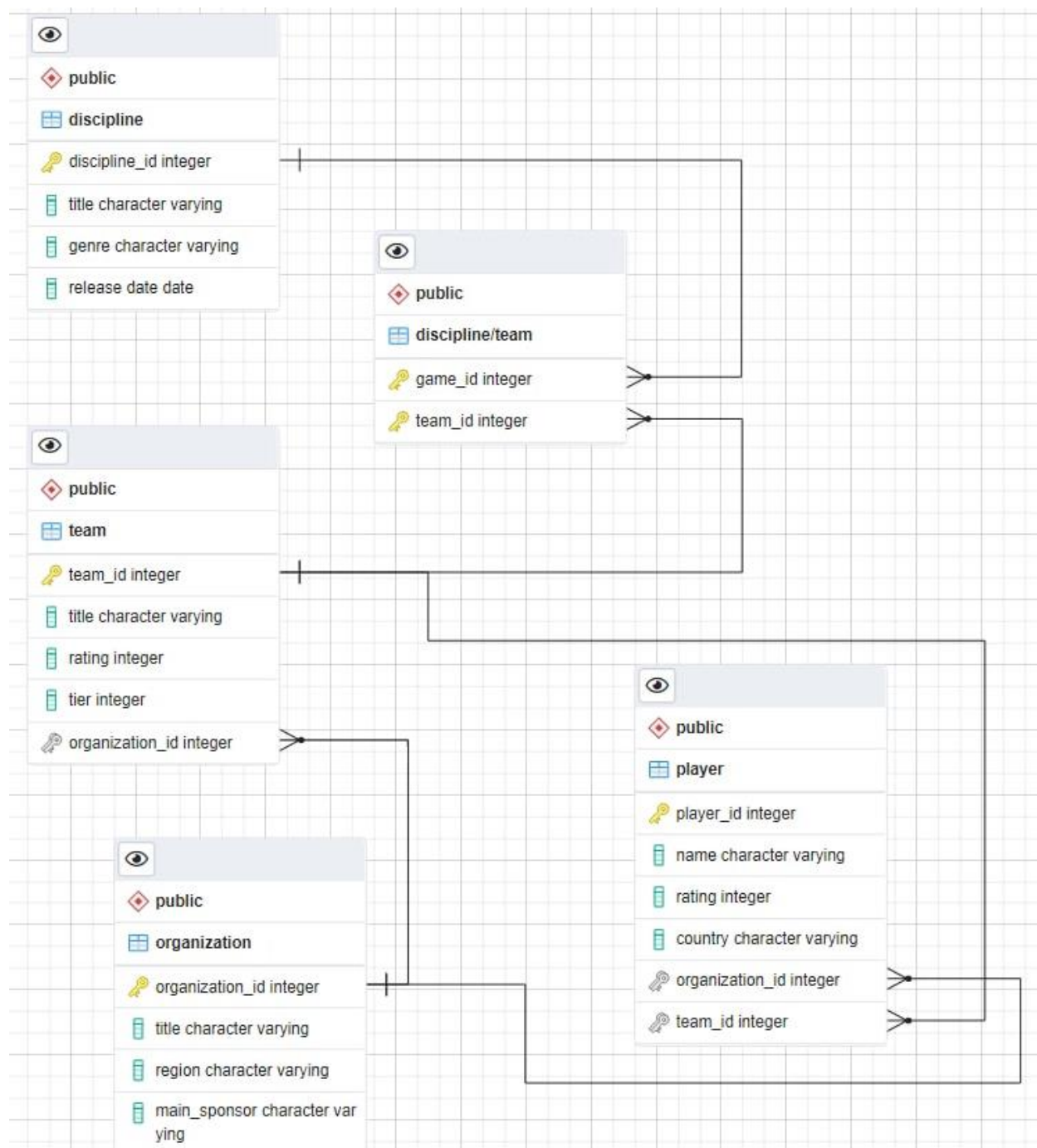
2. Якщо зв'язок типу N : M, то необхідно побудувати три таблиці - по одній для кожної сутності і одну для зв'язку. Первинний ключ сутності повинен бути первинним ключем відповідної таблиці. Таблиця для зв'язку серед своїх атрибутів повинна мати ключі обох сутностей. Даний зв'язок можна спостерігати між сутностями discipline та team.

Перетворена концептуальна модель у схему бази даних зображена на малюнку 2.



Мал. 2 - схема бази даних, побудована у додатку draw.io.

Після розробки моделі предметної галузі «Кіберспорт» та перетворення її у схему бази даних, було створено базу даних в додатку pgAdmin 4, схема представлена на малюнку 3.



Мал. 3 - Схема бази даних у додатку pgAdmin 4.

Опис структури бази даних

Сутність gaming discipline перетворена у таблицю gaming discipline, в ній присутній тільки один первинний ключ та відсутні зовнішні ключі. Сутність organization перетворено у таблицю organization, де теж тільки один первинний ключ та немає зовнішніх ключів. Сутність team перетворена у таблицю team, у цієї таблиці тільки один первинний ключ та за наявності зв'язків 1 : N маємо такий зовнішній ключ як organization_id. Сутність player перетворена у таблицю player, в даному випадку знаходиться тільки один первинний ключ та за наявності зв'язку 1 : N ще 2 зовнішні ключі: organization_id та team_id. Оскільки сутності gaming discipline та team зв'язані багато до багатьох, в нас з'явиться третя таблиця game/team. Вона має два первинних ключі: game_id та team_id. Опис даних сутностей та їх атрибутів наведено в таблиці 1.

Табл. 1 – опис структури Бази Даних.

Відношення	Атрибут	Тип атрибуту
gaming discipline – містить дані про ігрову дисципліну.	game_id – унікальний ідентифікатор title – назва гри genre – жанр release_date – офіційна дата виходу	integer (числовий) character varying (рядок) character varying (рядок) date (дата)
organization – містить дані про кіберспортивну організацію.	organization_id – унікальний ідентифікатор title – назва region – основний регіон організації main_sponsor – основний спонсор	integer (числовий) character varying (рядок) character varying (рядок) character varying (рядок)

team – містить дані про професійні команди	team_id – унікальний ідентифікатор title – назва команди rating – місце в рейтингу команд tier – рівень команди (1 – найвищий)	integer (числовий) character varying (рядок) integer (числовий) integer (числовий)
players – містить дані про гравців	player_id - унікальний ідентифікатор name – ім'я гравця rating – місце в рейтингу гравців country – рідна країна гравця	integer (числовий) character varying (рядок) integer (числовий) character varying (рядок)
game/team – відношення дисциплін до команд	game_id – ідентифікатор дисципліни team_id – ідентифікатор команди	integer (числовий) integer (числовий)

Нормалізація схеми бази даних до третьої нормальної форми

Функціональні залежності для кожної таблиці

1) Gaming discipline:

- game_id -> title (назва ігрової дисципліни залежить від її унікального ідентифікатора)
- game_id -> genre (жанр конкретної гри визначається за унікальним дисципліни)
- game_id -> release_date (дата виходу певної гри теж визначається за унікальним ідентифікатором)

2) organization:

- organization_id -> title (назва організації залежить від унікального ідентифікатора)

- b) organization_id -> region (конкретний регіон дізнається за унікальним ідентифікатором організації)
- c) organization_id -> main_sponsor (вся інформація про основного спонсора доступна завдяки унікальному ідентифікатору)

3) team:

- a) team_id -> title (назва команди залежить від унікального ідентифікатора)
- b) team_id -> rating (рейтинг конкретної команди залежить від її унікального ідентифікатора)
- c) team_id -> tier (рівень команди передається її унікальним ідентифікатором)

4) players:

- a) player_id -> name (ім'я гравця залежить від унікального ідентифікатора)
- b) player_id -> rating (рейтинг конкретного гравця залежить від його унікального ідентифікатора)
- c) player_id -> country (рідна країна гравця визначається його унікальним ідентифікатором)

Відповідність схеми бази даних до третьої нормальної форми

Схема відповідає 1НФ, тому що:

1. В таблиці немає дубльованих рядків.
2. В кожній комірці зберігається атомарне (скалярне) значення.
3. В кожному стовпці зберігаються дані одного типу.

Схема відповідає 2НФ, тому що:

1. Вона відповідає 1НФ.
2. Має первинний ключ, а всі не ключові стовпці таблиці залежать від первинного ключа.

Схема відповідає 3НФ, тому що:

1. Вона відповідає 2НФ.
2. Всі не ключові атрибути таблиці залежать винятково від усього первинного ключа, а не його частини. Тобто кожен неключовий атрибут нетранзитивно (без посередника) залежить від первинного ключа.

SQL текст

```
-- Table: public.discipline

-- DROP TABLE public.discipline;

CREATE TABLE IF NOT EXISTS public.discipline
(
    discipline_id integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1
MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),
    title character varying COLLATE pg_catalog."default" NOT NULL,
    genre character varying COLLATE pg_catalog."default" NOT NULL,
    "release date" date NOT NULL,
    CONSTRAINT discipline_pkey PRIMARY KEY (discipline_id)
)

TABLESPACE pg_default;

ALTER TABLE public.discipline
    OWNER to postgres;

-- Table: public.discipline/team

-- DROP TABLE public."discipline/team";

CREATE TABLE IF NOT EXISTS public."discipline/team"
(
    game_id integer NOT NULL,
    team_id integer NOT NULL,
    CONSTRAINT "discipline/team_pkey" PRIMARY KEY (game_id, team_id),
    CONSTRAINT fk_game_id FOREIGN KEY (team_id)
        REFERENCES public.team (team_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT fk_team_id FOREIGN KEY (game_id)
```

```

REFERENCES public.discipline (discipline_id) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION
NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE public."discipline/team"
    OWNER to postgres;

-- Table: public.organization

-- DROP TABLE public.organization;

CREATE TABLE IF NOT EXISTS public.organization
(
    organization_id integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START
1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),
    title character varying COLLATE pg_catalog."default" NOT NULL,
    region character varying COLLATE pg_catalog."default" NOT NULL,
    main_sponsor character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT organization_pkey PRIMARY KEY (organization_id)
)

TABLESPACE pg_default;

ALTER TABLE public.organization
    OWNER to postgres;

-- Table: public.player

-- DROP TABLE public.player;

CREATE TABLE IF NOT EXISTS public.player
(

```

```

    player_id integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1
MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),
    name character varying COLLATE pg_catalog."default" NOT NULL,
    rating integer NOT NULL,
    country character varying COLLATE pg_catalog."default" NOT NULL,
    organization_id integer NOT NULL,
    team_id integer NOT NULL,
    CONSTRAINT player_pkey PRIMARY KEY (player_id),
    CONSTRAINT fk_organization_id FOREIGN KEY (organization_id)
        REFERENCES public.organization (organization_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT fk_team_id FOREIGN KEY (team_id)
        REFERENCES public.team (team_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
)

```

```

TABLESPACE pg_default;

```

```

ALTER TABLE public.player
    OWNER to postgres;

```

```

-- Table: public.team

```

```

-- DROP TABLE public.team;

```

```

CREATE TABLE IF NOT EXISTS public.team

```

```

(
    team_id integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1
MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),
    title character varying COLLATE pg_catalog."default" NOT NULL,
    rating integer NOT NULL,
    tier integer NOT NULL,

```

```
organization_id integer NOT NULL,  
CONSTRAINT team_pkey PRIMARY KEY (team_id),  
CONSTRAINT fk_organization_id FOREIGN KEY (organization_id)  
    REFERENCES public.organization (organization_id) MATCH SIMPLE  
    ON UPDATE NO ACTION  
    ON DELETE NO ACTION  
    NOT VALID  
)  
  
TABLESPACE pg_default;  
  
ALTER TABLE public.team  
    OWNER to postgres;
```

Таблиці бази даних у pgAdmin 4

Gaming discipline

discipline

↗

✕

General

Columns

Advanced

Constraints

Parameters

Security

SQL

Inherited from table(s)

Select to inherit from...

Columns

+

		Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
<div><div>✎</div><div>🗑</div></div>		discipline_id	integer			<div>Yes</div>	<div>Yes</div>
<div><div>✎</div><div>🗑</div></div>		title	character varying			<div>Yes</div>	<div>No</div>
<div><div>✎</div><div>🗑</div></div>		genre	character varying			<div>Yes</div>	<div>No</div>
<div><div>✎</div><div>🗑</div></div>		release date	date			<div>Yes</div>	<div>No</div>

discipline

↗

✕

General

Columns

Advanced

Constraints

Parameters

Security

SQL

Primary Key

Foreign Key

Check

Unique

Exclude

+

		Name	Columns
<div><div>✎</div><div>🗑</div></div>		discipline_pkey	discipline_id

discipline

↗

✕

General

Columns

Advanced

Constraints

Parameters

Security

SQL

Primary Key

Foreign Key

Check

Unique

Exclude

+

		Name	Columns	Referenced Table
--	--	------	---------	------------------

Organization

organization

General

Columns

Advanced

Constraints

Parameters

Security

SQL

Inherited from table(s)

Select to inherit from...

Columns

		Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
		organization_id	integer			<div>Yes</div>	<div>Yes</div>
		title	character varying			<div>Yes</div>	<div>No</div>
		region	character varying			<div>Yes</div>	<div>No</div>
		main_sponsor	character varying			<div>Yes</div>	<div>No</div>

organization

General

Columns

Advanced

Constraints

Parameters

Security

SQL

Primary Key

Foreign Key

Check

Unique

Exclude

		Name	Columns
		organization_pkey	organization_id

organization

General

Columns

Advanced

Constraints

Parameters

Security

SQL

Primary Key

Foreign Key

Check

Unique

Exclude

		Name	Columns	Referenced Table
--	--	------	---------	------------------

Team

team

General

Columns

Advanced

Constraints

Parameters

Security

SQL

Inherited from table(s)

Select to inherit from...

Columns

		Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
		team_id	integer			<div>Yes</div>	<div>Yes</div>
		title	character varying			<div>Yes</div>	<div>No</div>
		rating	integer			<div>Yes</div>	<div>No</div>
		tier	integer			<div>Yes</div>	<div>No</div>
		organization_id	integer			<div>Yes</div>	<div>No</div>

team

General

Columns

Advanced

Constraints

Parameters

Security

SQL

Primary Key

Foreign Key

Check

Unique

Exclude

		Name	Columns
		team_pkey	team_id

team

General

Columns

Advanced

Constraints

Parameters

Security

SQL

Primary Key

Foreign Key

Check

Unique

Exclude

		Name	Columns	Referenced Table
		fk_organization_id	(organization_id) -> (organization_id)	public.organization

Player

player

General

Columns

Advanced

Constraints

Parameters













Security

SQL

Inherited from table(s)

Select to inherit from...

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
 	player_id	integer			<div>Yes</div>	<div>Yes</div>
 	name	character varying			<div>Yes</div>	<div>No</div>
 	rating	integer			<div>Yes</div>	<div>No</div>
 	country	character varying			<div>Yes</div>	<div>No</div>
 	organization_id	integer			<div>Yes</div>	<div>No</div>
 	team_id	integer			<div>Yes</div>	<div>No</div>

player

General

Columns

Advanced

Constraints

Parameters

Security

SQL


Primary Key

Foreign Key

Check

Unique

Exclude

	Name	Columns
 	player_pkey	player_id

player

General

Columns

Advanced

Constraints

Parameters

Security

SQL





Primary Key

Foreign Key

Check

Unique

Exclude

	Name	Columns	Referenced Table
 	fk_organization_id	(organization_id) -> (organization_id)	public.organization
 	fk_team_id	(team_id) -> (team_id)	public.team

Discipline/team

discipline/team

General

Columns

Advanced

Constraints

Parameters

Security

SQL

Inherited from table(s)

Select to inherit from...

Columns

game_id

integer

Yes

Yes

team_id

integer

Yes

Yes

discipline/team

GeneralColumnsAdvancedConstraintsParametersSecuritySQL

Primary KeyForeign KeyCheckUniqueExclude

Name

discipline/team_pkey





Columns

game_id,team_id

discipline/team






[General](#)
[Columns](#)
[Advanced](#)
[Constraints](#)
[Parameters](#)
[Security](#)
[SQL](#)

[Primary Key](#)
[Foreign Key](#)
[Check](#)
[Unique](#)
[Exclude](#)

		Name	Columns	Referenced Table
		fk_game_id	(team_id) -> (team_id)	public.team
		fk_team_id	(game_id) -> (discipline_id)	public.discipline

Вміст таблиць бази даних

Data Output	Explain	Messages	Notifications
 discipline_id [PK] integer 	title character varying 	genre character varying 	release date date 
1	1	CS:GO	Shooter, 1st person
2	2	Dota 2	MOBA
3	3	Overwatch	Fantasy, Shooter
4	4	PUBG	Shooter, 1st/3rd person

Data Output	Explain	Messages	Notifications
 organization_id [PK] integer 	title character varying 	region character varying 	main_sponsor character varying 
1	1	Natus Vincere	CIS
2	2	Virtus.pro	CIS
3	3	Team Liquid	USA
4	4	Team Spirit	CIS

Data Output		Explain	Messages	Notifications		
	player_id [PK] integer 	name character varying 	rating integer 	country character varying 	organization_id integer 	team_id integer 
1	5	Alexander S1mple Kostiliev	1	Ukraine	1	8
2	6	Nikolay somedieyoung Tegza	2	Russia	4	11
3	7	Steve Stewie2k Johnson	6	USA	3	9
4	8	Denis electronic Morozov	5	Russia	1	8
5	9	Marek yekindar Bikowic	8	Cerbia	2	10

Data Output		Explain	Messages	Notifications
 team_id [PK] integer 	title character varying 	rating integer 	tier integer 	organization_id integer 
1	8	Natus Vincere	1	1
2	9	Team Liquid	5	1
3	10	Virtus.pro	4	1
4	11	Team Spirit	12	2