

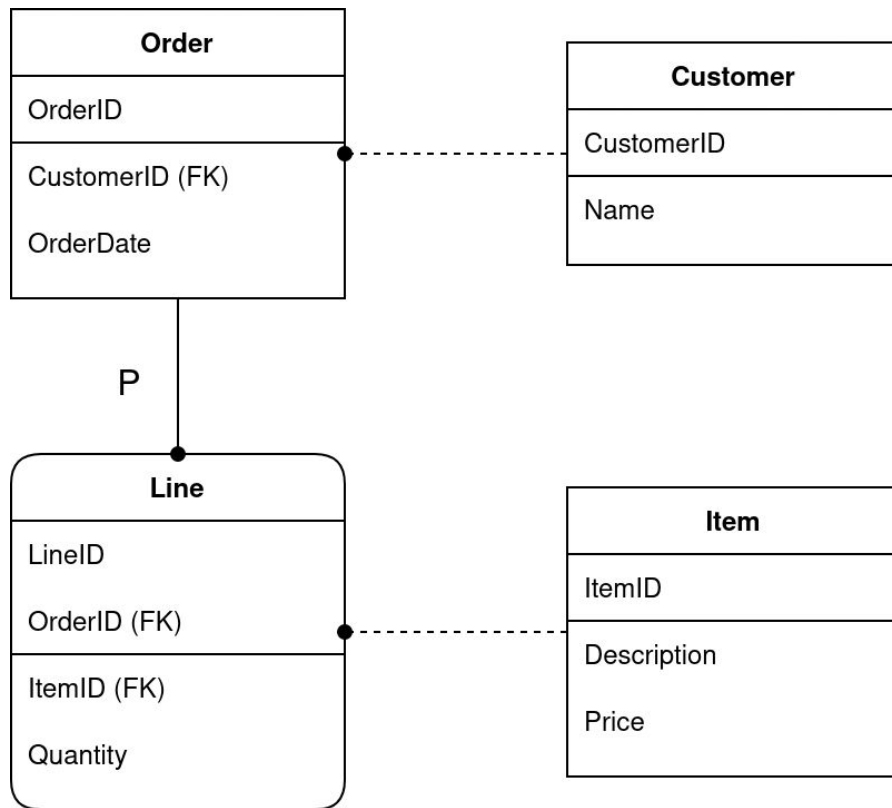


IDEF1X. PostgreSQL. psql.
pgAdmin.

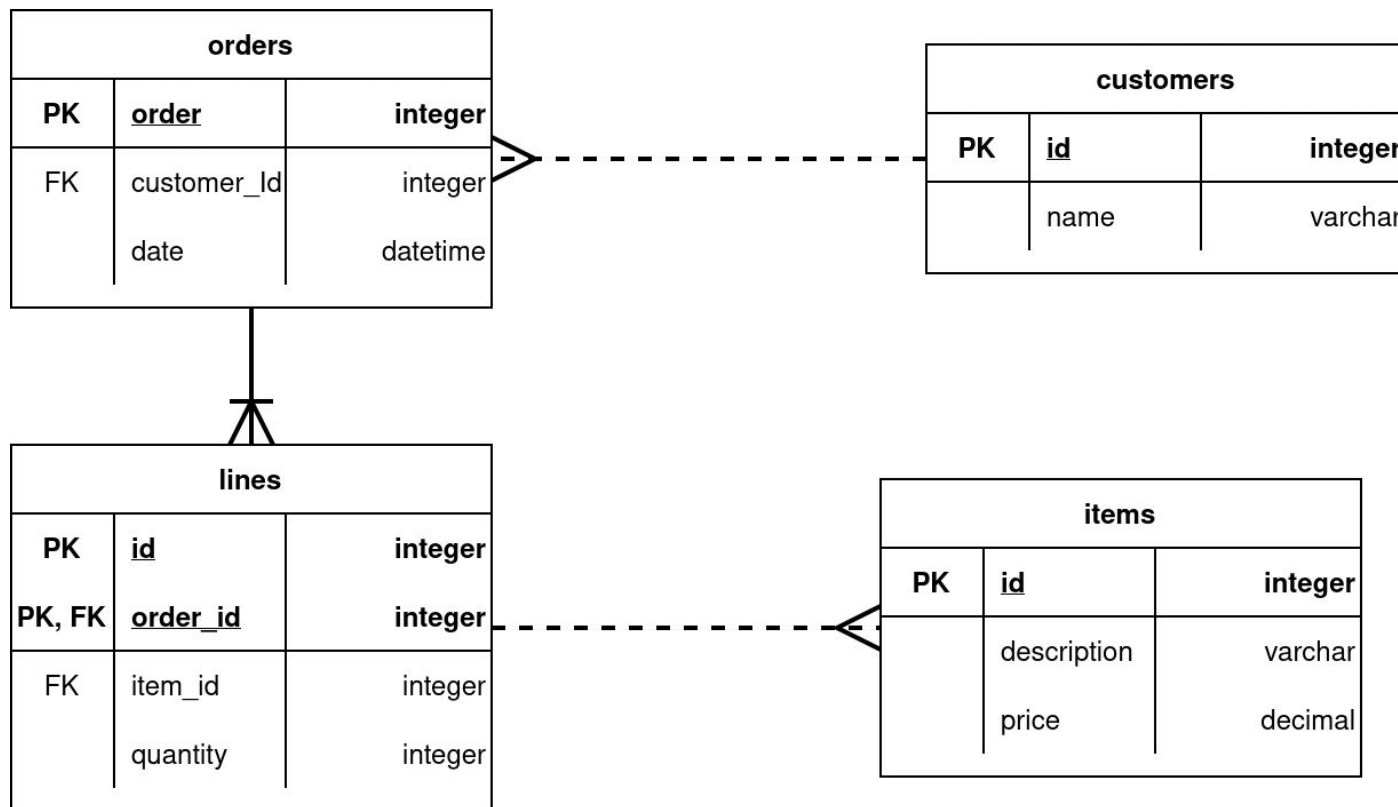
IDEF1X

Что такое IDEF1X?

Что такое IDEF1X?



Что такое IDEF1X?



Элементы построения модели данных

Независимая Сущность
Ключевые атрибуты
Неключевые атрибуты

Зависимая Сущность
Ключевые атрибуты
Неключевые атрибуты

Элементы построения модели данных

Независимая Сущность
Ключевые атрибуты
Неключевые атрибуты

Зависимая Сущность
Ключевые атрибуты
Неключевые атрибуты

0, 1 или больше



1 или больше



0 или 1



ровно N



идентифицирует



не идентифицирует



Элементы построения модели данных

Независимая Сущность
Ключевые атрибуты
Неключевые атрибуты

Зависимая Сущность
Ключевые атрибуты
Неключевые атрибуты

Таблица		
PK	<u>Ключевые атрибуты</u>	Тип данных
	Неключевые атрибуты	

0, 1 или больше ●—————

1 или больше ●P—————

0 или 1 ●Z—————

ровно N ●N—————

идентифицирует —————

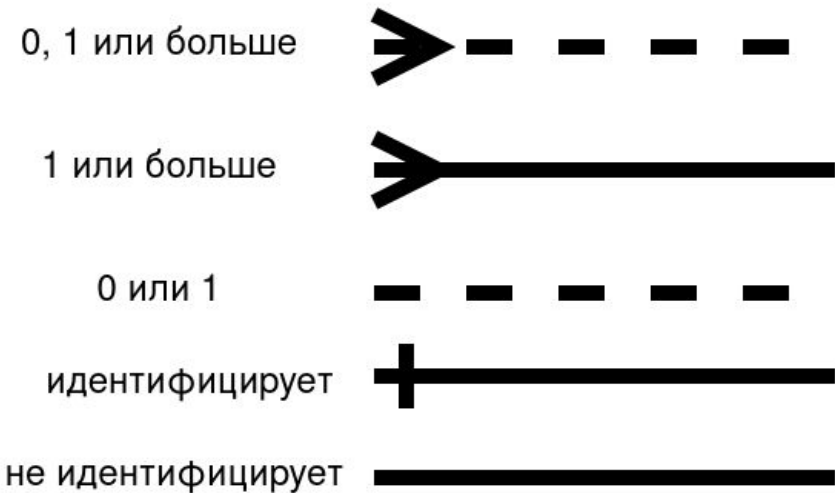
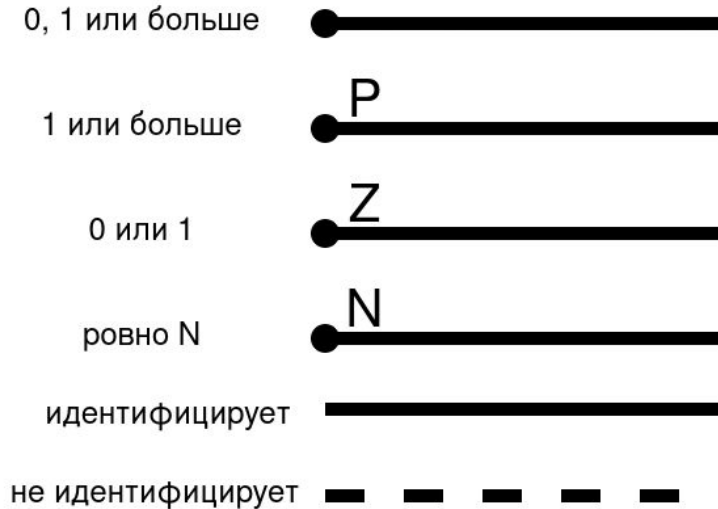
не идентифицирует - - - - -

Элементы построения модели данных

Независимая Сущность
Ключевые атрибуты
Неключевые атрибуты

Зависимая Сущность
Ключевые атрибуты
Неключевые атрибуты

Таблица		
PK	Ключевые атрибуты	Тип данных
	Неключевые атрибуты	



PostgreSQL

Установка

- [Оф. сайт, где можно найти ссылку на установщик/посмотреть нужные команды](#)
- [Windows и MacOS](#)
 - Порт лучше не менять
 - Локализацию ставить английскую
 - Сразу можно поставить pgAdmin
- Linux
 - `sudo curl https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -`
 - `sudo curl https://www.pgadmin.org/static/packages_pgadmin_org.pub | sudo apt-key add -`
 - `sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt $(lsb_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list'`
 - `sudo sh -c 'echo "deb https://ftp.postgresql.org/pub/pgadmin/pgadmin4/apt/$(lsb_release -cs) pgadmin4 main" > /etc/apt/sources.list.d/pgadmin4.list'`
 - `sudo apt update`
 - `sudo apt install postgresql postgresql-contrib pgadmin4`
 - `sudo usermod -a -G sudo postgres`
 - `sudo -u postgres initdb --locale en_US.UTF-8 -D '/var/lib/postgres/data'`
 - `sudo -u postgres pg_ctl -D /var/lib/postgres/data -l logfile start`

psql

- psql (psql.exe) --username postgres или sudo -U postgres psql должно сработать в большинстве случаев
- Параметры
 - -U <user> – зайти под определенным пользователем (по умолчанию пользователь, выполнивший команду)
 - -h <host> – зайти на определенный хост (по умолчанию localhost)
 - -d <dbname> – зайти на определенную бд (по умолчанию имя пользователя, выполнившего команду)
 - -p <port> – зайти на определенный порт (по умолчанию 5432)
 - -W – запросить пароль при входе (форсированно)
 - -u – запросить ввод логина и пароля при входе
 - -c <command> – выполнить определенную команду
 - -l – запросить список баз данных
 - [Полный список параметров](#)

Внутри psql – команды

- \l – список баз данных
- \c <dbname> – подключиться к определенной базе
- \dt *.* – список таблиц базы
- \du – список пользователей
- \d <table> – описание таблицы
- \q – выйти
- \h – помощь по синтаксису SQL
- \? – список доступных команд

Внутри psql – конфигурируем для использования

1. `CREATE ROLE <username> WITH LOGIN PASSWORD '<password>';` –
создаем пользователя

Внутри psql – конфигурируем для использования

1. `CREATE ROLE <username> WITH LOGIN PASSWORD '<password>';` – создаем пользователя
 - a. `ALTER ROLE <username> WITH SUPERUSER;` – если нужны права суперпользователя

Внутри psql – конфигурируем для использования

1. `CREATE ROLE <username> WITH LOGIN PASSWORD '<password>';` – создаем пользователя
 - a. `ALTER ROLE <username> WITH SUPERUSER;` – если нужны права суперпользователя
 - b. Аналогично можно добавлять другие атрибуты пользователя

Внутри psql – конфигурируем для использования

1. `CREATE ROLE <username> WITH LOGIN PASSWORD '<password>';` – создаем пользователя
 - a. `ALTER ROLE <username> WITH SUPERUSER;` – если нужны права суперпользователя
 - b. Аналогично можно добавлять другие атрибуты пользователя
2. `CREATE DATABASE <dbname> WITH OWNER <username>;` – создаем БД

Внутри psql – конфигурируем для использования

1. `CREATE ROLE <username> WITH LOGIN PASSWORD '<password>';` – создаем пользователя
 - a. `ALTER ROLE <username> WITH SUPERUSER;` – если нужны права суперпользователя
 - b. Аналогично можно добавлять другие атрибуты пользователя
2. `CREATE DATABASE <dbname> WITH OWNER <username>;` – создаем БД
3. `\c <dbname>` – переходим в созданную БД

Внутри psql – конфигурируем для использования

1. `CREATE ROLE <username> WITH LOGIN PASSWORD '<password>';` – создаем пользователя
 - a. `ALTER ROLE <username> WITH SUPERUSER;` – если нужны права суперпользователя
 - b. Аналогично можно добавлять другие атрибуты пользователя
2. `CREATE DATABASE <dbname> WITH OWNER <username>;` – создаем БД
3. `\c <dbname>` – переходим в созданную БД
4. ...

Внутри psql – конфигурируем для использования

1. `CREATE ROLE <username> WITH LOGIN PASSWORD '<password>';` – создаем пользователя
 - a. `ALTER ROLE <username> WITH SUPERUSER;` – если нужны права суперпользователя
 - b. Аналогично можно добавлять другие атрибуты пользователя
2. `CREATE DATABASE <dbname> WITH OWNER <username>;` – создаем БД
3. `\c <dbname>` – переходим в созданную БД
4. ...
5. Начинаем хулиганить

Внутри psql – конфигурируем для использования

1. `CREATE ROLE <username> WITH LOGIN PASSWORD '<password>';` – создаем пользователя
 - a. `ALTER ROLE <username> WITH SUPERUSER;` – если нужны права суперпользователя
 - b. Аналогично можно добавлять другие атрибуты пользователя
2. `CREATE DATABASE <dbname> WITH OWNER <username>;` – создаем БД
3. `\c <dbname>` – переходим в созданную БД
4. ...
5. Начинаем хулиганить

Вместо SQL-запросов для создания объектов БД можно использовать утилиты `createuser --interactive <username>` (`createuser.exe`) и `createdb -O <username> <dbname>` (`createdb.exe`) соответственно

PostgreSQL готовим для сервера

PostgreSQL готовим для сервера

1. `psql -c 'SHOW pg_hba;'` – покажет путь до файла `pg_hba.conf`

PostgreSQL готовим для сервера

1. `psql -c 'SHOW pg_hba;'` – покажет путь до файла `pg_hba.conf`
2. В этом файле добавляем строки
 - a. `local <dbname> <username> trust`
 - b. `host <dbname> <username> 0.0.0.0/0 md5`

PostgreSQL готовим для сервера

1. `psql -c 'SHOW pg_hba;'` – покажет путь до файла `pg_hba.conf`
2. В этом файле добавляем строки
 - a. `local <dbname> <username> trust`
 - b. `host <dbname> <username> 0.0.0.0/0 md5`
3. `psql -c 'SELECT pg_reload_conf();'` – перечитает файлы настроек (конфигурацию)

PostgreSQL готовим для сервера

1. `psql -c 'SHOW pg_hba;'` – покажет путь до файла `pg_hba.conf`
2. В этом файле добавляем строки
 - a. `local <dbname> <username> trust`
 - b. `host <dbname> <username> 0.0.0.0/0 md5`
3. `psql -c 'SELECT pg_reload_conf();'` – перечитает файлы настроек (конфигурацию)

`pg_hba.conf` – файл конфигурации, в котором настраиваются доступы к СУБД по таким атрибутам как база данных, пользователь, IP-запроса.

pgAdmin

- Как psql, только с GUI'ем
- Имеет возможность работать в psql
- Можно смотреть схемы БД
- Real time мониторинг метрик
- Генерация DDL, DML, DCL команд для выполняемых действий
- ERD-диаграммы и генерация DDL команд с помощью них



- Databases (4)
 - oauth2_accounts
 - Casts
 - Catalogs (2)
 - Event Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas (1)
 - public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Tables (13)
 - access_tokens
 - accounts
 - alembic_version
 - clients
 - clients_codes
 - fingerprints
 - links
 - log_tables
 - passwords
 - refresh_tokens
 - sessions_data
 - social_accounts
 - socials
 - Trigger Functions
 - Types
 - Views

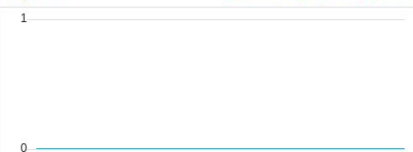
Database sessions



Transactions per second



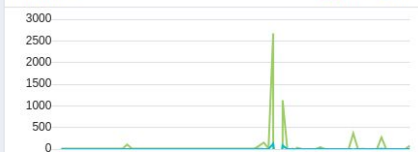
Tuples in



Tuples out



Block I/O



Server activity

Sessions

Locks

Prepared Transactions



Search

			PID	User	Application	Client	Backend start	Transaction start	State	Wait event	Blocking PIDs
✖	■	▶	2236087	postgres	pgAdmin 4 - DB:oauth2_a...	::1	2022-04-12 11:46:30 +05	2022-04-12 11:46:57 +05	active		

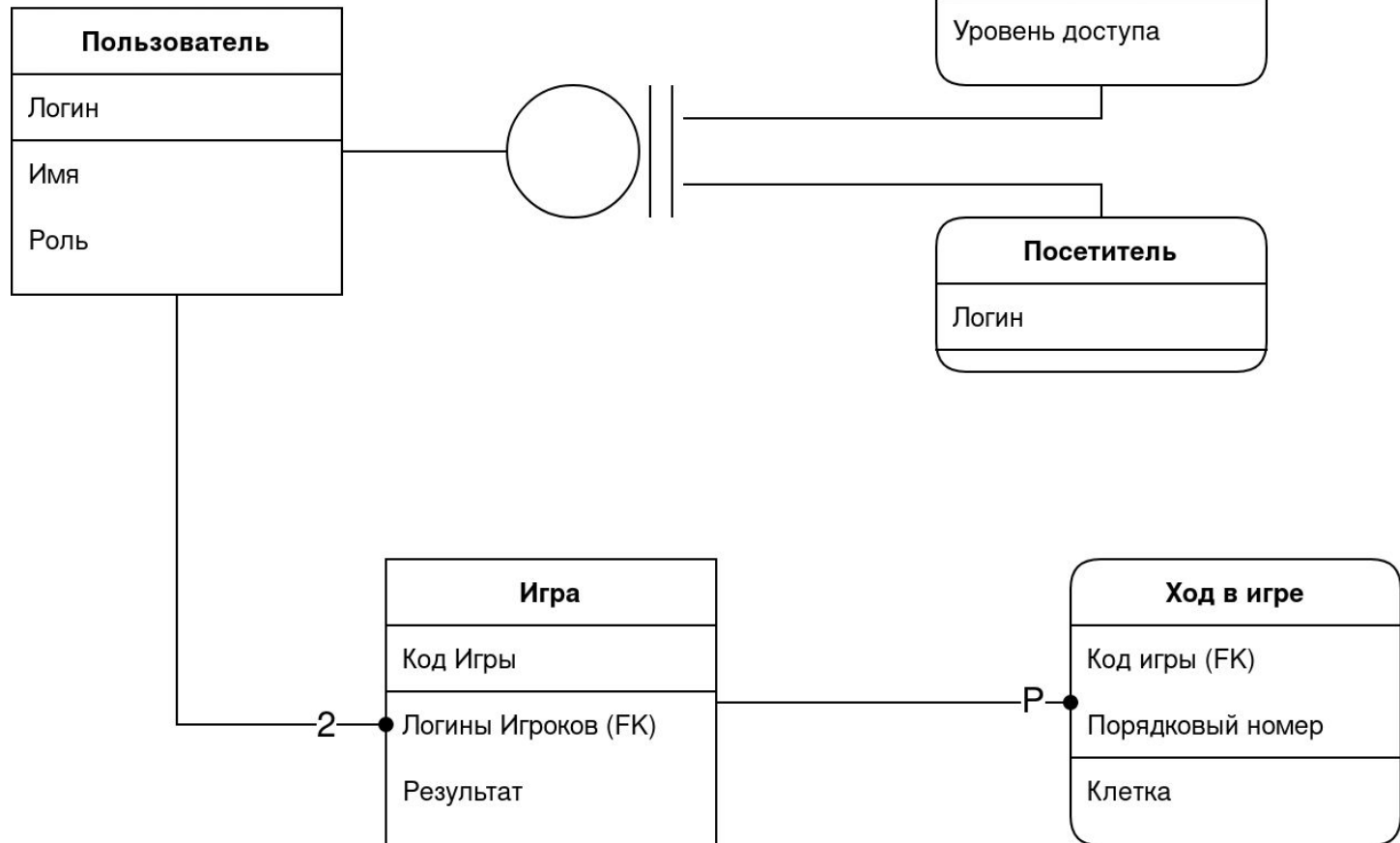
Задание

- Развернуть PostgreSQL
- Создать таблицы с занятия
- Добавить внешние ключи, которые имеются в модели
- Заполнить таблицы данными (минимум 5 пользователей, минимум 5 игр)
- Написать запрос, который бы собирал информацию о количестве побед/поражений/ничьих пользователя
- Написать запрос, который бы собирал полную статистику игр пользователя
- Написать запрос, который бы выводил топ-5 игроков по соотношению побед/поражений/ничьих
- Переделать таблицу turns так, чтобы избавиться от id хода



Справочный материал

- [PostgreSQL документация](#)
- [Шпаргалка по psql](#)
- [Рисовалка для схем](#)



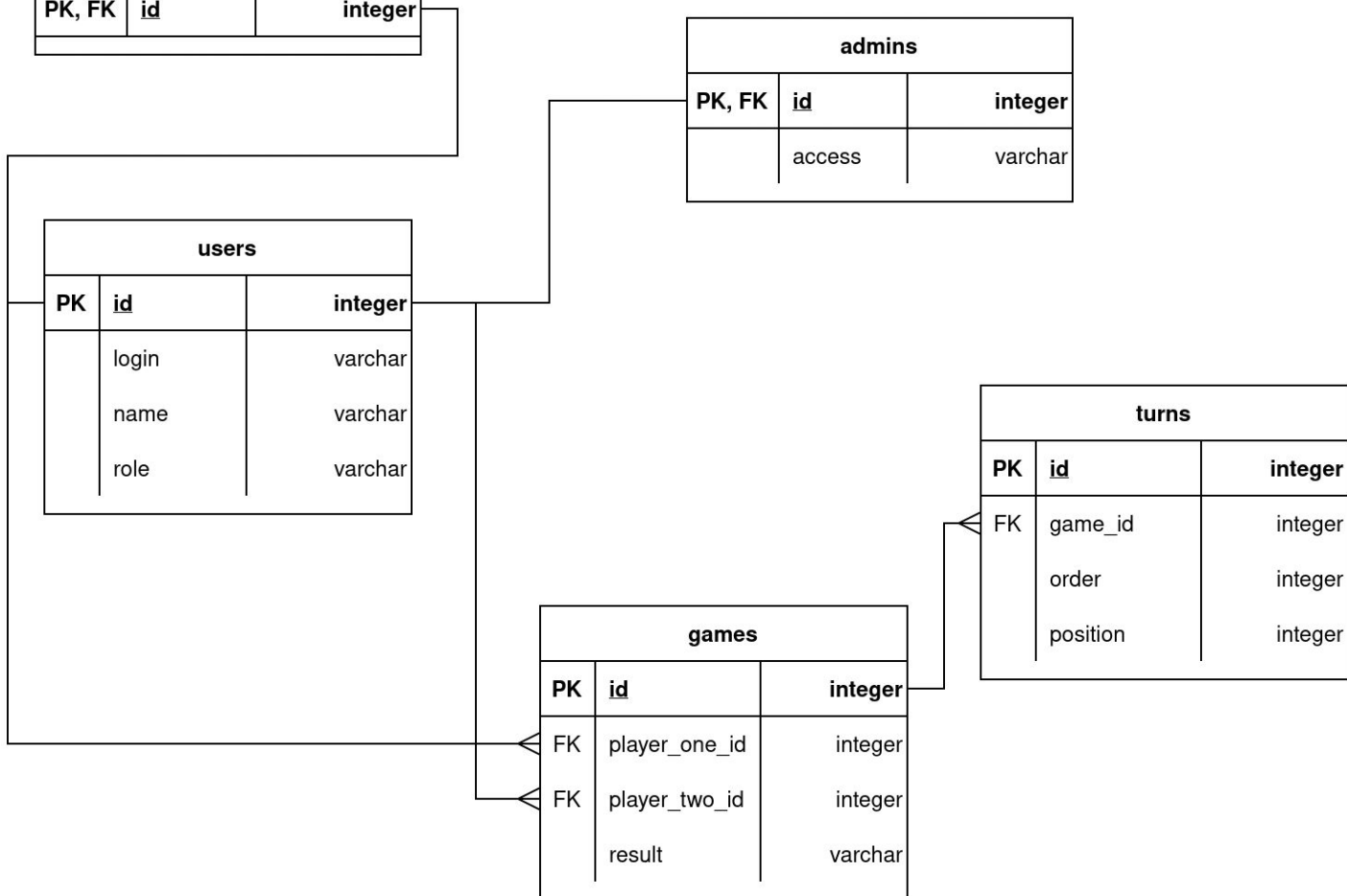
regulars		
PK, FK	<u>id</u>	integer

admins		
PK, FK	<u>id</u>	integer
	access	varchar

users		
PK	<u>id</u>	integer
	login	varchar
	name	varchar
	role	varchar

turns		
PK	<u>id</u>	integer
FK	game_id	integer
	order	integer
	position	integer

games		
PK	<u>id</u>	integer
FK	player_one_id	integer
FK	player_two_id	integer
	result	varchar





```
CREATE ROLE tictactoe WITH LOGIN PASSWORD 'Passw0rd';  
CREATE DATABASE tictactoe WITH OWNER tictactoe;
```

<https://pastebin.com/FrpV34xb>

<https://pastebin.com/Byse5qRx>



```
CREATE TABLE users(  
    id SERIAL PRIMARY KEY,  
    login VARCHAR UNIQUE not NULL,  
    name VARCHAR not NULL  
);  
  
CREATE TABLE games(  
    id SERIAL PRIMARY KEY,  
    player_one_id INTEGER NOT NULL,  
    player_two_id INTEGER NOT NULL,  
    result VARCHAR NULL  
);  
  
CREATE TABLE turns(  
    id SERIAL PRIMARY KEY,  
    game_id INTEGER NOT NULL,  
    "order" INTEGER NOT NULL,  
    position VARCHAR NULL  
);
```