

Модель КП соединения ПВО

Модель базы данных, реализующей функционал ведения боевых действий соединения ПВО (напр., зенитно-ракетного дивизиона).

Установка

Для работы требуется PostgreSQL 9.6+ с установленным расширением PostGIS 2.0+.

- [Установка PostgreSQL](#)
- [Установка PostGIS](#). На ОС семейства Debian пакет доступен в репозиториях APT. Пакет [поставляется](#) с ОС AstraLinux.

Запуск

1. Выполнить файлы в каталоге `setup` по очереди в соответствии с их номерами. Файлы должны быть выполнены от имени суперпользователя (напр., `postgres`). shell script `sudo -u postgres psql -f 1_initdb.sql sudo -u postgres psql -f 2_objects.sql`
2. Далее возможны варианты действий:
 - Запустить файл `operations/example.sh` от имени суперпользователя `shell script sudo -u postgres operations/example.sh`
 - Выполнить SQL-файлы в каталоге `operations` по очереди в соответствии с их номерами. При выполнении файла `fr24data.sql` для каждой транзакции вызывается набор триггеров, которые могут приводить к пуску ракеты.

Файлы

Каталог `setup`

После выполнения файлов в данном каталоге модель готова к использованию.

`initdb.sql`

Создаёт расширение PostGIS и задаёт три роли для пользователей модели.

`objects.sql`

Создаёт объекты модели (таблицы, функции, представления, триггеры).

Каталог `operations`

Включает файлы для демонстрации работы модели.

`user.sql`

Создаёт учётные записи пользователей.

Используются учётные записи `petrov (missile_launcher)`, `sidorov (viewer)` и `radar (radar_data_provider)` с паролями, равными их именам.

`self.sql`

Задаёт положение ЗРК, ЗУР и обороняемых объектов.

`fr24data.sql`

Содержит запросы для вставки "первичных" радиолокационных данных.

`parse_fr24_data.py`

Программа на Python3 для конвертации треков полётов воздушных судов из CSV-формата FlightRadar24 в SQL-запросы.

Таблицы и представления БД

В БД входит несколько таблиц и представлений. Все они включены в схему `cc`.

Первичные радиолокационные данные должны поступать в таблицу `radar_data`.

Для получения информации о состоянии системы используются следующие представления: `* objects_with_radar_data`. Выявленные воздушные цели и их радиолокационные данные. `* objects_near_defended_objects`, `objects_near_sam_systems`. Выявленные воздушные цели в радиусе 1,5 от обороняемых объектов и ЗРК. `* launched_missiles`. Запущенные ЗУР.

Данные о собственных объектах хранятся в следующих таблицах: `* defended_objects`. Обороняемые объекты. `* sam_systems`. ЗРК. `* missiles`. ЗУР.

Выявленные цели хранятся в таблице `objects`.