**Текстовое описание базы данных**

Данная база данных создана для функционирования предприятия. Ее основная задача – поддерживать актуальную информацию о процессах, происходящих в компании. Компания работает в области воздушных беспилотных технологий (дроны, беспилотники и т.д.).

**База данных состоит из 11 таблиц:**

* Раздел проектов компании:

1. Projects – описывает текущие проекты компании

* Раздел штата компании:

1. ODOP – список сотрудников – диспетчеров, занимающихся организацией безопасности полетов.

Для того, чтобы в таблице не было двух диспетчеров с повторяющимися именами, фамилиями, отчествами и датами рождения, реализован уникальный индекс.

1. Operators – список сотрудников – операторов, занимающихся полетами. Для того, чтобы в таблице не было двух операторов с повторяющимися именами, фамилиями, отчествами и датами рождения, реализован уникальный индекс.
2. Head\_of\_field\_party – список руководителей полевых партий для каждого проекта, которые выбираются из списка операторов.

* Раздел оборудования компании:

1. UAS\_type – раздел, содержащий модели БВС, присутствующих в компании.
2. UAS – таблица, содержащая список всех БВС компании, а также оффсеты к ним. Для того, чтобы в таблице не было задвоения БВС, реализован уникальный индекс на основе регистрационного номера БВС и даты приобретения БВС.

* Раздел обеспечения безопасности полетов:

1. Reg\_centers – раздел, содержащий список региональных центров, а также ссылок на шаблоны документов для каждого регионального центра.
2. Regims – таблица, описывающая архив установленных диспетчерами режимов. Для того, чтобы избежать задвоения добавленных режимов, реализован уникальный индекс по типу и номеру режима.

* Производственный раздел компании:

1. Flights – таблица, содержащая список всех полетов компании. Для того, чтобы избежать задвоения полетов, реализован уникальный индекс по дате взлета, посадки, а также операторам, реализующим данный полет.
2. Operator\_bonus – таблица, содержащая информацию о премии каждого оператора за каждый месяц. Для того, чтобы оператору не начислили премию дважды за месяц, реализован уникальный индекс по id оператора и дате расчета.
3. History\_offsets – таблица типа Archive, содержащая историю изменений оффсетов.

Для воплощения в жизнь данной базы данных необходим WEB – интерфейс, написанный на Django. Центральная часть базы данных – таблица flights, описывающая все произведенные операторами полеты. Данные в таблице появляются после того, как оператор совершил полет и загрузил логи полета через WEB – интерфейс. Цепляется следующая информация:

- Дата и время начала и конца полета

- Продолжительность

- Ссылка на папку с логами полета

- Проект, согласно которому выполнялся полет

- Список операторов, выполнявших полет

- БВС, которым выполнялся полет

- Режим использования воздушного пространства, согласно которому выполнялся полет.

Можно сделать вывод о том, что для корректного заполнения таблицы flights, требуется информация о всех беспилотниках (БВС), имеющихся в компании. Данную информацию можно взять из таблицы UAS. Также для того, чтобы полет был выполнен в соответствии с правилами безопасности выполнения полетов, необходимо, чтобы полет находился в режиме использования воздушного пространства. Данные режимы устанавливаются диспетчера путем направления в соответствующие региональные центры запросов, шаблоны которых содержаться в данной базе данных (таблица Reg\_centers). Установленные режимы хранятся в таблице Regims.

Возникает необходимость в расчете полетного времени, от которого зависит расчет премии для операторов (таблица Operators\_bonus). Данное время раcсчитывается как разность времени взлета и посадки, которые считываются из логов полета на этапе загрузки данных на сервер. Для реализации данных расчетов необходимо реализовать скрипт на Python.

Сохранение логов полета в отдельной папке для каждого полета предназначено для дальнейшей автоматической проверки данных логов скриптами Python на появление предупреждений / опасных ситуаций и других явлений, способных повлечь ущерб для компании и окружающей среды, во время полета с бортом. Для проверки логов необходимо реализовать скрипт на Python.

Также, в таблице UAS содержится актуальная информация об оффсетах для каждого БВС, которые необходимы для дальнейшей камеральной обработки материалов. Зная БВС, которые выполнял данный полет, можно подтянуть из данной таблицы оффсеты.

В ходе курсового проекта был реализован триггер, отвечающий за создание резервной копии всех оффсетов (при обновлении этих оффсетов) из таблицы UAS в таблице History\_offsets. База данных резервной копии реализована специально на движке Archive, чтобы не было возможности удалить / изменить данные.

Также в ходе курсового проекта были реализованы три представления:

1. get\_names\_of\_heads\_of\_fields\_party - Представление для отображения имен руководителей партий.
2. who\_is\_his\_head - Представление для отображения имени руководителя партии для каждого оператора.
3. how\_much\_employees\_for\_every\_head - Представление для отображения количества сотрудников у каждого руководителя полевой партии.