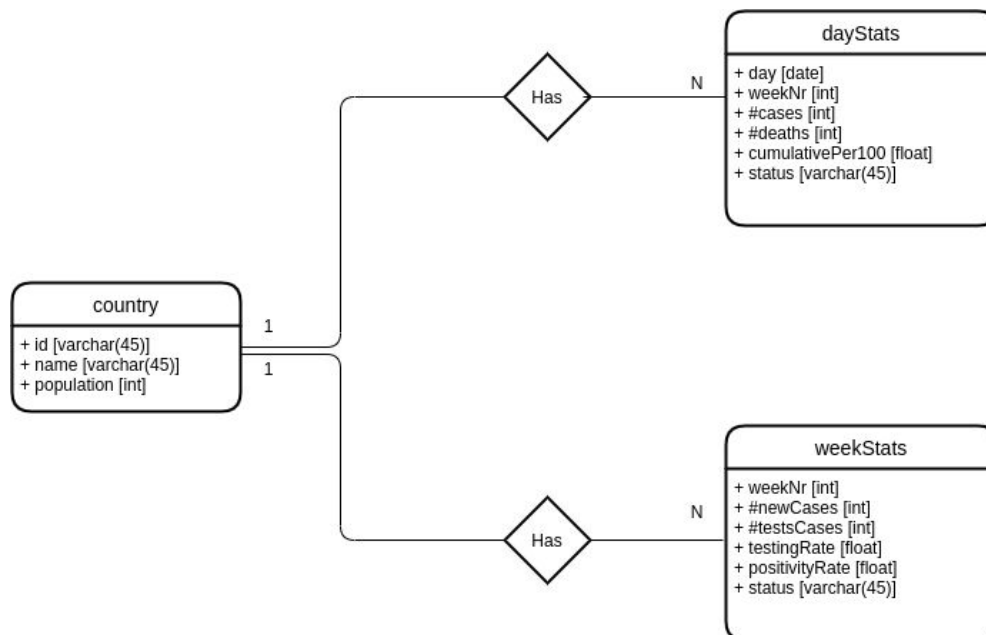


Análise de Dados - TP01

Vasco Ramos, PG42852

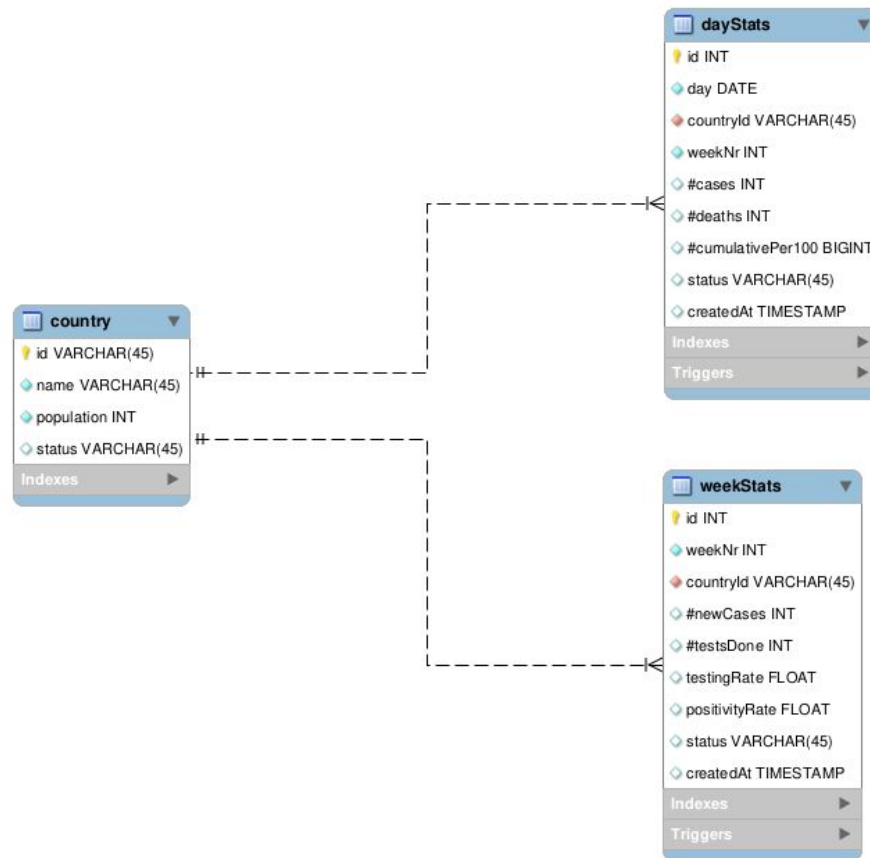
Criação do Modelo Conceptual e EER

A abordagem inicial foi tentar mapear os dados fornecidos para um esquema conceptual, onde acabei por encontrar 3 entidades: **País**, **Estatísticas Diárias** e **Estatísticas Semanais**:



Como se pode ver na imagem, cada país tem a sua informação básica (o seu GeoCode como identificador, nome e população) e associado a si um N número de estatísticas associadas, tanto diárias como semanais. Para além disto, as outras duas tabelas, são uma extração “natural” dos dois ficheiros CSV fornecidos.

O passo seguinte foi converter este Modelo Conceptual para o respetivo ERR, que, através do *MySQL Workbench* ficou com o seguinte desenho:



Criação do Modelo Físico

Após terminar o EER, e novamente através do *MySQL Workbench*, procedi à criação das tabelas, dos *triggers* e de um *stored procedure*.

O *stored procedure* foi criado para permitir um *Soft Delete* de entradas, para ser possível manter um histórico de ações, como se pode ver:

```
create procedure softDeleteEntry (
    tableName varchar(64),
    id varchar(64)
)
begin
    set @tableName = tableName;
    set @id = id;
    set @sqlText = concat('update ', @tableName, ' set status=\deleted\ where id=\'', @id, '\');

    prepare stmt from @sqlText;
    execute stmt;
    deallocate prepare stmt;
end
```

Por outro lado, os *triggers* foram criados para permitir uma associação automática de um timestamp de criação a todas as entradas de estatísticas, quer diárias como semanais:

```
CREATE DEFINER = CURRENT_USER TRIGGER `covid`.`dayStats_init_created_at` BEFORE INSERT ON `dayStats` FOR EACH ROW
BEGIN
    SET NEW.createdAt = now();
END

CREATE DEFINER = CURRENT_USER TRIGGER `covid`.`weekStats_init_created_at` BEFORE INSERT ON `weekStats` FOR EACH ROW
BEGIN
    SET NEW.createdAt = now();
END
```

É também de ressaltar que tanto os *triggers*, como o *stored procedure* foram criados através e com o apoio do *MySQL Workbench*.

Povoação das tabelas

O passo seguinte foi povoar as tabelas. Para isso desenvolvi um script em Python que gera um ficheiro SQL com todos os inserts necessários para a Base de Dados. Na imagem é possível ver a estrutura principal do programa:

```
def main():
    with open("COVID19_EU_EEA_UK_DATA.csv") as csv_file:
        csv_reader = csv.reader(csv_file, delimiter=";")

        count = 0

        for row in csv_reader:
            if count != 0:
                insert_country(row)
                insert_day_stats(row)

            count += 1

    with open("COVID19_EU_EEA_UK_TESTING.csv") as csv_file:
        csv_reader = csv.reader(csv_file, delimiter=";")

        count = 0

        for row in csv_reader:
            if count != 0:
                insert_week_stats(row)

            count += 1

    with open("populate_sript.sql", "w") as script_file:
        script_file.write("use covid;")
        script_file.write(countries)
        script_file.write(day_stats)
        script_file.write(week_stats)
```

Queries à BD

O passo final era interrogar a Base de Dados com algumas *queries*. As duas queries iniciais foram usadas para relembrar os conceitos mais básicos, sendo que a primeira lista todas as estatísticas diárias da 43^a semana em Portugal e a segunda lista todas as estatísticas semanais de Portugal.

A terceira, e última, query é um pouco mais avançada e tratar de calcular e listar para todos os países a taxa de mortalidade, tendo em conta o número total de casos e mortos de que há registo.