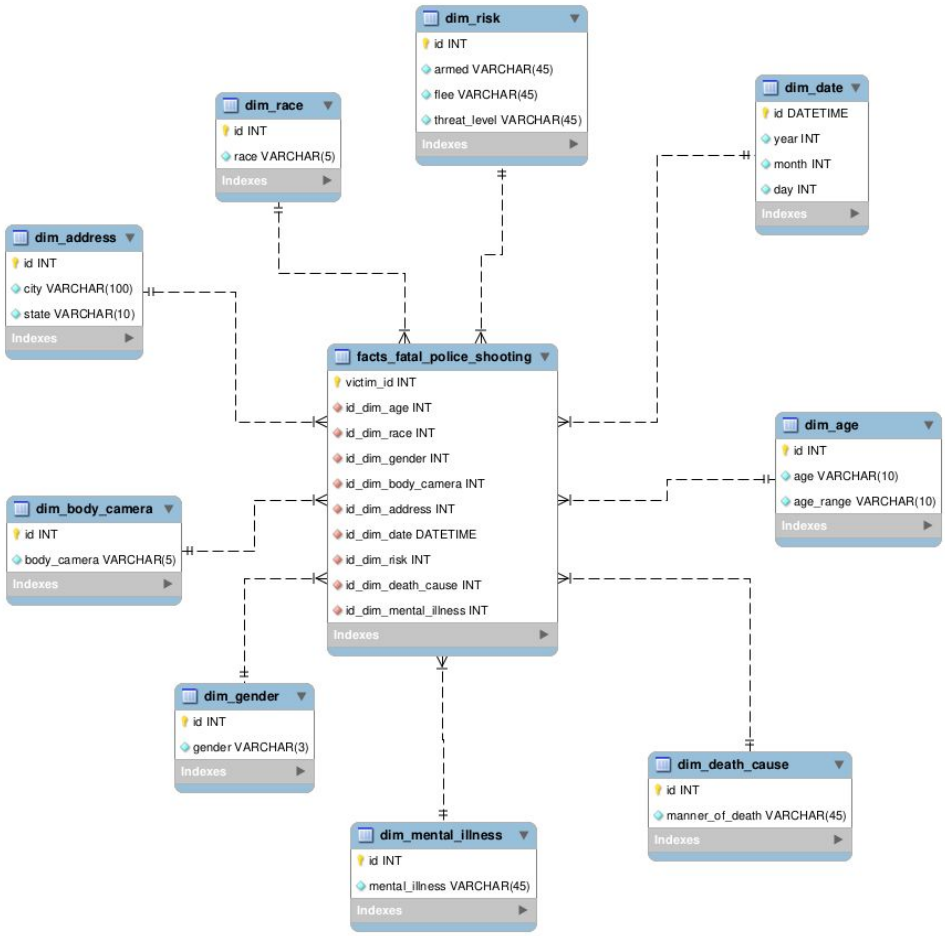
Análise de Dados - TP03

Vasco Ramos, PG42852

Criação do EER (Modelo Dimensional em Estrela)

Para criar um modelo lógico cujo modelo dimensional fosse um esquema de estrela, a partir do CSV fornecido, o primeiro passo foi perceber qual era a entidade da tabela de facto (neste caso, o incidente de morte por tiro) e a partir daí tentar agrupar colunas do CSV representativas de conceitos correlacionáveis em tabelas de dimensão associadas à tabela de facto. O resultado final foi o seguinte modelo:

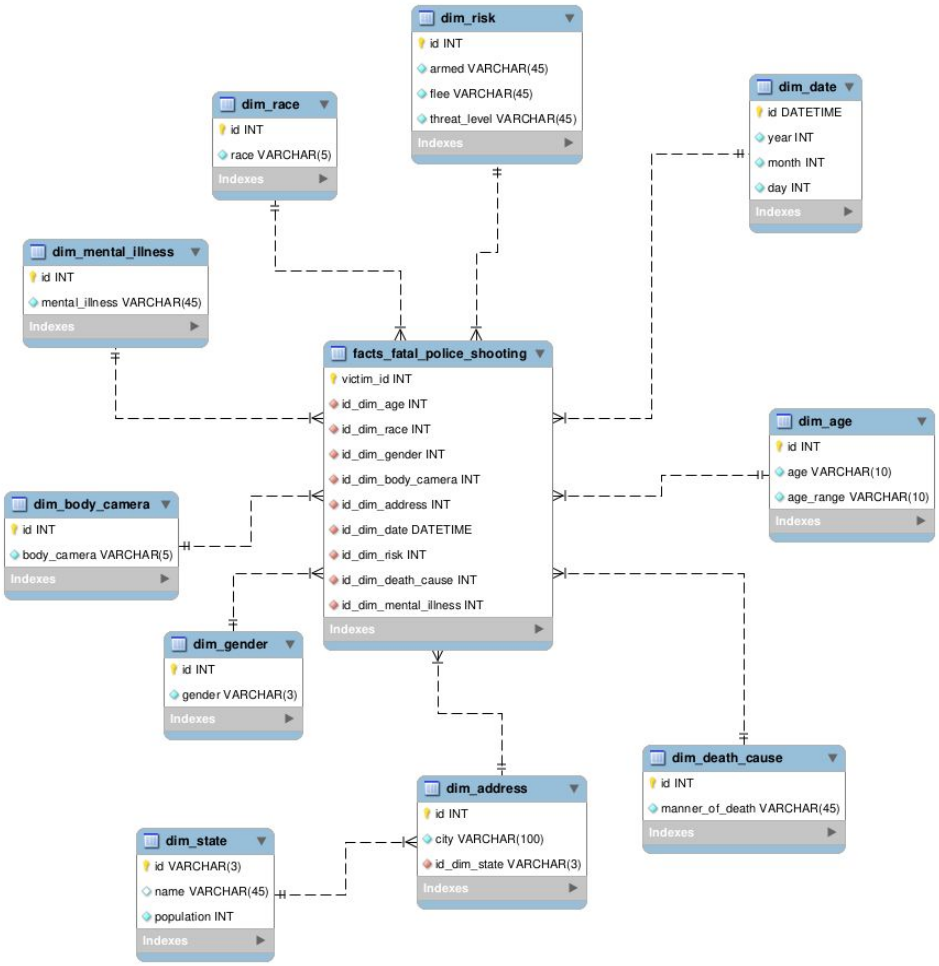


As tabelas dimensionais de **idade**, **sexo**, **raça** e **localização** são relativamente autoexplicativas. Decidi também criar uma tabela de dimensão para modelar o **eixo tempo**, um pouco para ser possível perceber uma evolução temporal do número de casos e perceber se havia alguma correlação com certos períodos na vida dos Estados Unidos, nomeadamente

períodos de maior fração e divisão social, conflitos ideológicos, etc. E, noutra vertente, achei por bem criar outras três tabelas de dimensão: **risco**, presença de **câmara corporal** e **causa de morte**, pois, achei que estes três eixos eram essenciais para melhor compreender estes fenómenos. Nas tabelas de câmara corporal e causa de morte apenas consegui encontrar um atributo para cada uma delas, contudo, na tabela de risco achei que fazia sentido relacionar: se a vítima estava armada, se estava a fugir e o nível de ameaça que representava, na visão do agente da polícia.

Criação do EER (Modelo Dimensional em Floco de Neve)

Este modelo ficou sensivelmente semelhante ao anterior com uma grande diferença de incluir a informação de cada estado (nome completo, abreviatura e população) como um segundo nível de informação, o que originou, então, o esquema em floco de neve. Isto, porque achei que fazia sentido explorar a vertente de análise de número de casos por estado e também relacionar esses números com a densidade populacional dos mesmos para conseguirmos ter os números em perspetiva. Assim sendo, esta abordagem deu origem ao seguinte esquema em floco de neve:



Criação do Modelo Físico

Nos dois modelos, o passo da criação do modelo físico foi feito através do *MySQL Workbench*.

Povoação das tabelas

Após a criação do modelo físico, passei para o povoamento das tabelas. Ao contrário dos últimos exercícios, não usei um script em *Python*, optei por fazer tudo através do *MySQL Workbench*, começando por inserir os dados do CSV numa tabela de dados temporária e, depois, criar as tabelas efetivas através de queries a essa tabela temporária, como pode ser visto na seguinte imagem:

```sql
-- populate dim_race
insert into dim_race (race)
select distinct race from temp_dataset;

-- populate dim_age
insert into dim_age (age, age_range)
select distinct age,
    case
        when age between 0 and 9 then '0s'
        when age between 10 and 19 then '10s'
        when age between 20 and 29 then '20s'
        when age between 30 and 39 then '30s'
        when age between 40 and 49 then '40s'
        when age between 50 and 59 then '50s'
        when age between 60 and 69 then '60s'
        when age between 70 and 79 then '70s'
        when age between 80 and 89 then '80s'
        when age between 90 and 99 then '90s'
        when age between 100 and 109 then '100s'
        else 'N/A'
    end as 'age_range'
from temp_dataset;

-- populate dim_gender
insert into dim_gender (gender)
select distinct gender from temp_dataset;

-- populate dim_body_camera
insert into dim_body_camera (body_camera)
select distinct body_camera from temp_dataset;
```

Queries à BD

O passo final era interrogar a Base de Dados com um conjunto de *queries*, neste caso foram cinco:

- Listar os 5 estados com maior número de incidentes.
- List o número de incidentes em cada cidade, por raça e sexo.
- Listar a percentagem de incidentes tendo em conta se os agentes têm ou não câmara corporal.
- Listar o número de casos com correlação ao nível do risco.
- Verificar a evolução temporal do número de casos.

Na imagem abaixo, fica a representação das queries listadas em código SQL:

```sql
use shootings_star;

-- top 5 states with more number of incidents
select state, count(victim_id) as `#incidents`
from facts_fatal_police_shooting as f
    join dim_address as da on f.id_dim_address = da.id
group by state
order by `#incidents` desc
limit 5;

-- list the number of incidents in each city, per race and sex
select city, race, gender, count(victim_id) as `#incidents`
from facts_fatal_police_shooting as f
    join dim_address as da on f.id_dim_address = da.id
    join dim_race as dr on f.id_dim_race = dr.id
    join dim_gender as dg on f.id_dim_gender = dg.id
group by city, race, gender
order by `#incidents` desc;

-- see the percentage of incidents that occur depending on whether the police officer has the body camera on or not
select body_camera, (count(victim_id) / temp.total_incidents) * 100 as `incidents (%)`
from facts_fatal_police_shooting as f
    join dim_body_camera as dbc on f.id_dim_body_camera = dbc.id
    join (select count(*) as `total_incidents` from facts_fatal_police_shooting) as temp
group by body_camera, temp.total_incidents;

-- list the number of cases with corellation with (the victim was armed, fleeing, or presented a high level of threat to the officer)
select armed, flee, threat_level, count(victim_id) as `#incidents`
from facts_fatal_police_shooting as f
    join dim_risk as dr on f.id_dim_risk = dr.id
group by armed, flee, threat_level
order by `#incidents` desc;

-- check the evolution of the number of shootings over the last years, taking the rising level of
-- indivualism and polarity present in the USA (discrimnation, xenophobia, ...)
select `year`, `month`,  count(victim_id) as `#incidents`
from facts_fatal_police_shooting as f
    join dim_date as dd on f.id_dim_date = dd.id
group by `year`, `month`
order by `year`, `month` asc;
```