

QRY03 - ChatGPT

User

Como melhorar a query: SELECT namebasics.nconst, namebasics.primaryname, namebasics.birthyear, namebasics.deathyear, titleprincipals.tconst, titlebasics.primarytitle FROM namebasics INNER JOIN titleprincipals ON titleprincipals.nconst = namebasics.nconst INNER JOIN titlebasics ON titlebasics.tconst = titleprincipals.tconst INNER JOIN titleakas ON titleakas.titleid = titlebasics.tconst WHERE namebasics.birthyear is not null and namebasics.deathyear is null and titleakas.isoriginaltitle = '1' order by namebasics.nconst

Com o schema:

```
CREATE TABLE public.namebasics (  
    index bigint,  
    nconst text,  
    primaryname text,  
    birthyear text,  
    deathyear text,  
    primaryprofession text,  
    knownfortitles text  
);  
ALTER TABLE public.namebasics OWNER TO postgres;  
CREATE TABLE public.titleakas (  
    index bigint,  
    titleid text,  
    ordering bigint,  
    title text,  
    region text,  
    language text,  
    types text,  
    attributes text,  
    isoriginaltitle text  
);  
ALTER TABLE public.titleakas OWNER TO postgres;  
CREATE TABLE public.titlebasics (  
    index bigint,  
    tconst text,  
    titletype text,  
    primarytitle text,  
    originaltitle text,  
    isadult text,  
    startyear text,  
    endyear text,
```

```

    "runtimeMinutes" text,
    genres text
);
ALTER TABLE public.titlebasics OWNER TO postgres;
CREATE TABLE public.titlecrew (
    index bigint,
    tconst text,
    directors text,
    writers text
);
ALTER TABLE public.titlecrew OWNER TO postgres;
CREATE TABLE public.titleepisode (
    index bigint,
    tconst text,
    parenttconst text,
    seasonnumber text,
    episodenummer text
);
ALTER TABLE public.titleepisode OWNER TO postgres;
CREATE TABLE public.titleprincipals (
    index bigint,
    tconst text,
    ordering bigint,
    nconst text,
    category text,
    job text,
    characters text
);
ALTER TABLE public.titleprincipals OWNER TO postgres;
CREATE TABLE public.titleratings (
    index bigint,
    tconst text,
    averagerating double precision,
    numvotes bigint
);
ALTER TABLE public.titleratings OWNER TO postgres;
CREATE INDEX ix_namebasics_index ON public.namebasics USING btree (index);
CREATE INDEX ix_titleakas_index ON public.titleakas USING btree (index);
CREATE INDEX ix_titlebasics_index ON public.titlebasics USING btree (index);
CREATE INDEX ix_titlecrew_index ON public.titlecrew USING btree (index);
CREATE INDEX ix_titleepisode_index ON public.titleepisode USING btree (index);
CREATE INDEX ix_titleprincipals_index ON public.titleprincipals USING btree (index);
CREATE INDEX ix_titleratings_index ON public.titleratings USING btree (index);
CREATE INDEX namebasics_pk ON public.namebasics USING btree (nconst COLLATE "C"
bpchar_pattern_ops);
CREATE INDEX titleakas_pk ON public.titleakas USING btree (titleid COLLATE "C"
bpchar_pattern_ops, ordering);
CREATE INDEX titlebasics_pk ON public.titlebasics USING btree (tconst COLLATE "C"

```

```
bpchar_pattern_ops);  
CREATE INDEX titlecrew_pk ON public.titlecrew USING btree (tconst COLLATE "C"  
bpchar_pattern_ops);  
CREATE INDEX titleepisode_pk ON public.titleepisode USING btree (tconst COLLATE "C"  
bpchar_pattern_ops);  
CREATE INDEX titleprincipals_pk ON public.titleprincipals USING btree (tconst COLLATE "C"  
bpchar_pattern_ops, ordering);  
CREATE INDEX titleratings_pk ON public.titleratings USING btree (tconst COLLATE "C"  
bpchar_pattern_ops);
```

ChatGPT

Sua consulta SQL parece eficiente e bem estruturada, mas sempre há espaço para melhorias.

Algumas sugestões para otimizar sua consulta:

Índices Compostos:

Certifique-se de que existam índices compostos para as colunas usadas nas cláusulas WHERE e JOIN. Isso pode acelerar a execução da consulta. Adicione índices compostos nas colunas usadas em condições de junção e filtragem.

```
CREATE INDEX idx_namebasics_birthdeath ON public.namebasics (birthyear, deathyear);  
CREATE INDEX idx_titleprincipals_nconst ON public.titleprincipals (nconst);  
CREATE INDEX idx_titlebasics_tconst ON public.titlebasics (tconst);  
CREATE INDEX idx_titleakas_titleid ON public.titleakas (titleid, isoriginaltitle);
```

Evite Uso de Funções em Cláusulas WHERE:

Evite usar funções ou manipulações em colunas nas cláusulas WHERE, pois isso pode impedir o uso eficiente de índices. Se possível, armazene os dados de data como tipos de data e evite o uso de `NULL` para representar anos desconhecidos.

Usar Alias para Melhor Legibilidade:

Use aliases (apelidos) para tornar a consulta mais legível. Isso é especialmente útil quando

you have several tables and need to refer to specific columns.

```
SELECT nb.nconst, nb.primaryname, nb.birthyear, nb.deathyear, tp.tconst, tb.primarytitle
FROM namebasics AS nb
INNER JOIN titleprincipals AS tp ON tp.nconst = nb.nconst
INNER JOIN titlebasics AS tb ON tb.tconst = tp.tconst
INNER JOIN titleakas AS ta ON ta.titleid = tb.tconst
WHERE nb.birthyear IS NOT NULL AND nb.deathyear IS NULL AND ta.isoriginaltitle = '1'
ORDER BY nb.nconst;
```

Rever o Uso de Índices e Estatísticas:

Periodicamente, revise a eficácia dos índices e atualize as estatísticas para garantir que o otimizador de consulta tome decisões informadas.

Lembre-se de testar essas alterações em um ambiente controlado antes de implementá-las em produção, pois o desempenho real pode variar dependendo do volume de dados e do ambiente do banco de dados.