

Como melhorar a query: select \* from titlebasics where titlebasics.tconst in (select titleepisode.tconst from titleepisode where titleepisode.parenttconst = 'tt0108778' order by titleepisode.seasonnumber, titleepisode.episodenumber)

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

Para melhorar a consulta, você pode considerar usar uma junção explícita em vez de uma subconsulta na cláusula `IN`. Isso muitas vezes melhora o desempenho da consulta. Aqui está uma versão otimizada da sua consulta:

```
SELECT tb.*
FROM titlebasics tb
JOIN titleepisode te ON tb.tconst = te.tconst
WHERE te.parenttconst = 'tt0108778'
ORDER BY te.seasonnumber, te.episodenumber;
```

Nesta consulta:

- Substituí a subconsulta na cláusula `IN` por uma junção explícita usando a cláusula `JOIN`.
- Usei aliases de tabela (`tb` e `te`) para tornar a consulta mais legível.
- Movemos as condições de junção para a cláusula `ON` da junção, onde faz mais sentido.

Esta versão da consulta deve ser mais eficiente e mais fácil de entender. Além disso, se necessário, você pode adicionar índices às colunas usadas nas condições de junção (`tconst` e `parenttconst`) para melhorar ainda mais o desempenho, dependendo do volume de dados na tabela.

### User

Quais índices podem ser criados para melhorar o desempenho desta query ?

### ChatGPT

Para melhorar o desempenho da sua consulta, você pode considerar a criação de índices nas colunas utilizadas nas cláusulas de junção e nas condições do filtro. Neste caso, as colunas relevantes são `tconst` na tabela `titlebasics` e `tconst` e `parenttconst` na tabela `titleepisode`. Aqui estão os índices recomendados:

```
CREATE INDEX idx_titlebasics_tconst ON public.titlebasics(tconst);

CREATE INDEX idx_titleepisode_tconst ON public.titleepisode(tconst);
CREATE INDEX idx_titleepisode_parenttconst ON public.titleepisode(parenttconst);
```

Estes índices serão particularmente úteis para melhorar o desempenho da junção entre as tabelas `titlebasics` e `titleepisode`, bem como para a condição de filtro `te.parenttconst = 'tt0108778'`. Lembre-se de que a criação de índices adiciona algum overhead durante a atualização ou inserção de dados, então, ao decidir criar índices, leve em consideração o equilíbrio entre leitura e gravação em sua aplicação.

Após criar esses índices, avalie o desempenho da consulta para garantir que eles estão tendo o impacto desejado. Você pode usar ferramentas de perfilamento de consultas ou `EXPLAIN ANALYZE` para analisar o plano de execução da consulta e identificar eventuais gargalos.