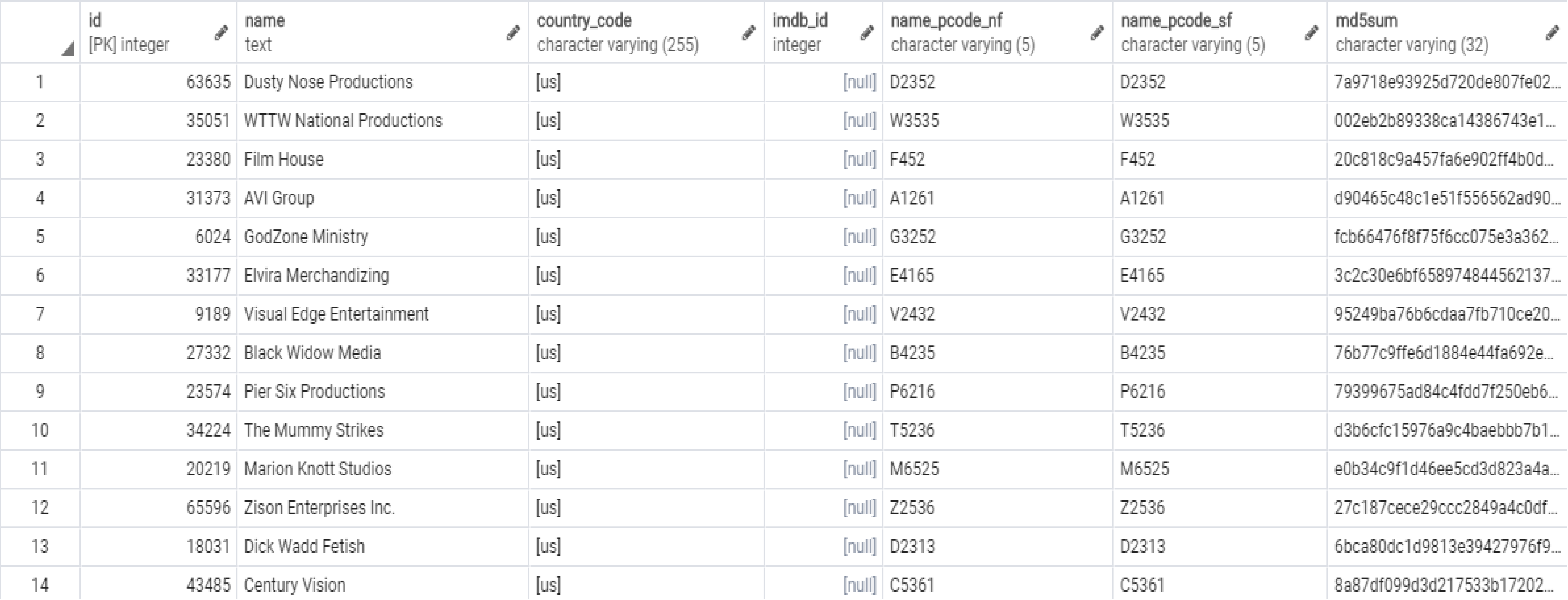
查询规模估算
Cardinality Estimation

2020/12/1

查询规模\查询基数估算



	id [PK] integer	name text	country_code character varying (255)	imdb_id integer	name_pcode_nf character varying (5)	name_pcode_sf character varying (5)	md5sum character varying (32)
1	63635	Dusty Nose Productions	[us]	[null]	D2352	D2352	7a9718e93925d720de807fe02…
2	35051	WTTW National Productions	[us]	[null]	W3535	W3535	002eb2b89338ca14386743e1…
3	23380	Film House	[us]	[null]	F452	F452	20c818c9a457fa6e902ff4b0d…
4	31373	AVI Group	[us]	[null]	A1261	A1261	d90465c48c1e51f556562ad90…
5	6024	GodZone Ministry	[us]	[null]	G3252	G3252	fcb66476f8f75f6cc075e3a362…
6	33177	Elvira Merchandizing	[us]	[null]	E4165	E4165	3c2c30e6bf658974844562137…
7	9189	Visual Edge Entertainment	[us]	[null]	V2432	V2432	95249ba76b6cdaa7fb710ce20…
8	27332	Black Widow Media	[us]	[null]	B4235	B4235	76b77c9ffe6d1884e44fa692e…
9	23574	Pier Six Productions	[us]	[null]	P6216	P6216	79399675ad84c4fdd7f250eb6…
10	34224	The Mummy Strikes	[us]	[null]	T5236	T5236	d3b6cfc15976a9c4baebbb7b1…
11	20219	Marion Knott Studios	[us]	[null]	M6525	M6525	e0b34c9f1d46ee5cd3d823a4a…
12	65596	Zison Enterprises Inc.	[us]	[null]	Z2536	Z2536	27c187cece29ccc2849a4c0df…
13	18031	Dick Wadd Fetish	[us]	[null]	D2313	D2313	6bca80dc1d9813e39427976f9…
14	43485	Century Vision	[us]	[null]	C5361	C5361	8a87df099d3d217533b17202…

更复杂的情况……

```
SELECT * FROM title  WHERE production_year > 1995
SELECT * FROM title  WHERE production_year > 2010
SELECT * FROM title  WHERE production_year > 2018
```

非均匀分布

```
SELECT * FROM title AS t
        WHERE t.production_year > 2005 AND revenues >100,000,000
```

属性相关

```
SELECT * FROM title AS t, movie_info_idx AS mi_idx
        WHERE t.production_year > 2005 AND   mi_idx.info < 8.5
        AND t.id = mi_idx.movie_id
```

JOIN

任务说明

https://www.kaggle.com/c/ruccardinality

根据SQL查询语句，预测查询规模

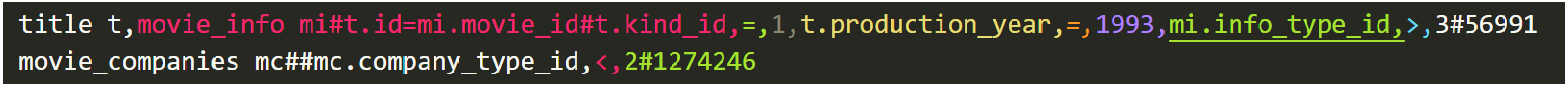
SQL类型

- 数值型数据（范围查询与等值查询）
- 最多涉及两表连接
- 提供查询计划（可选）

提供100000条训练集，需要预测5000条测试集的结果

数据解释

Training_data.csv
Testing_data.csv



```
title t,movie_info mi#t.id=mi.movie_id#t.kind_id,=,1,t.production_year,=,1993,mi.info_type_id,>,3#56991
movie_companies mc##mc.company_type_id,<,2#1274246
```

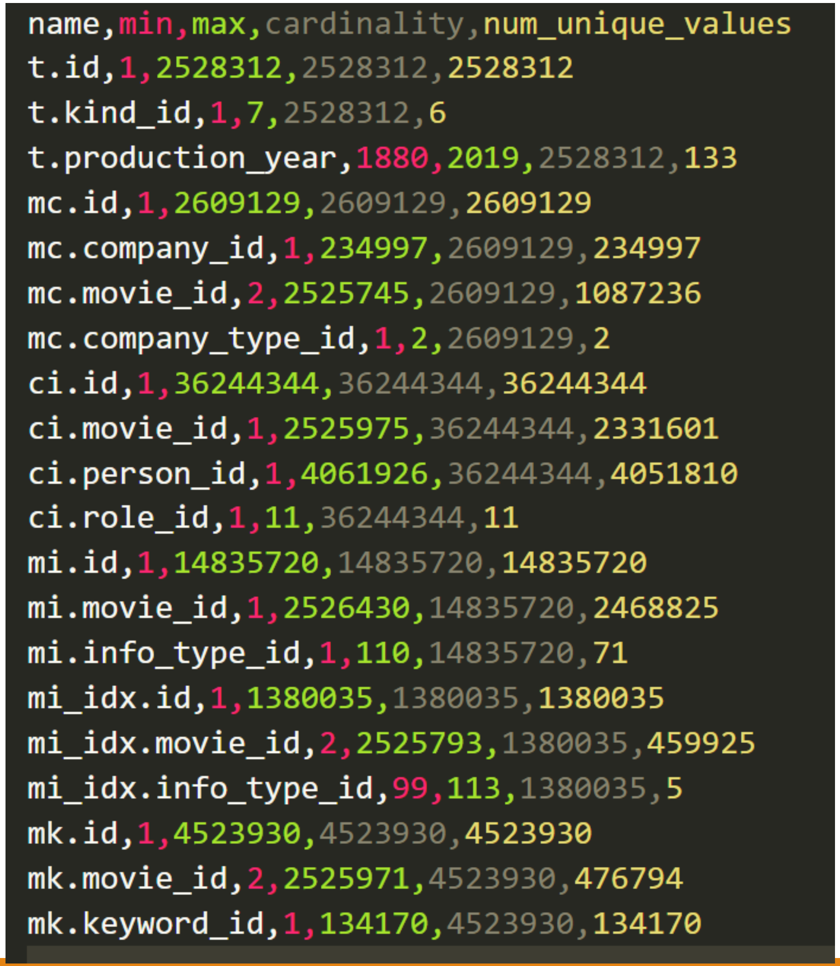
SELECT * FROM title t, movie_keyword mk WHERE mk.keyword_id<1029 AND t.id=mk.movie_id

title t,movie_keyword mk # t.id=mk.movie_id # mk.keyword_id,<,1029 #1038381

SELECT * FROM title t WHERE t.production_year >2005

title t # # t.production_year,>,2005

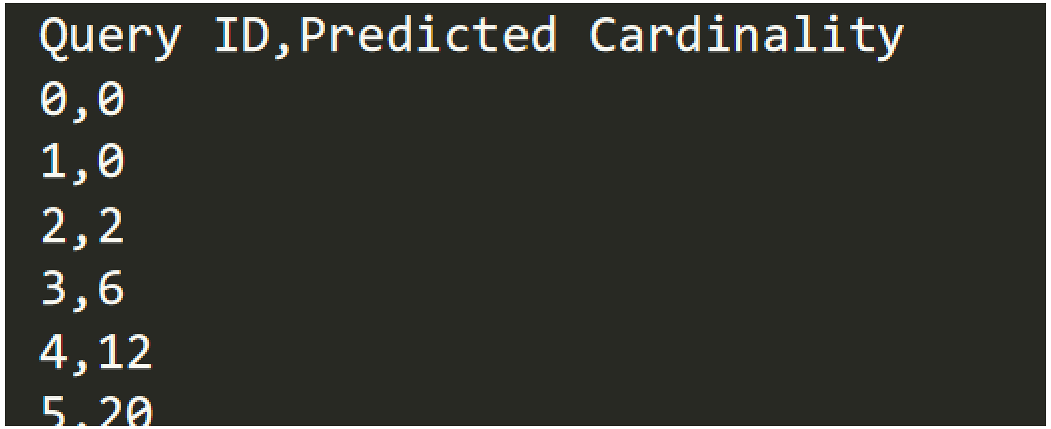
Column_min_max_vals.csv



```
name,min,max,cardinality,num_unique_values
t.id,1,2528312,2528312,2528312
t.kind_id,1,7,2528312,6
t.production_year,1880,2019,2528312,133
mc.id,1,2609129,2609129,2609129
mc.company_id,1,234997,2609129,234997
mc.movie_id,2,2525745,2609129,1087236
mc.company_type_id,1,2,2609129,2
ci.id,1,36244344,36244344,36244344
ci.movie_id,1,2525975,36244344,2331601
ci.person_id,1,4061926,36244344,4051810
ci.role_id,1,11,36244344,11
mi.id,1,14835720,14835720,14835720
mi.movie_id,1,2526430,14835720,2468825
mi.info_type_id,1,110,14835720,71
mi_idx.id,1,1380035,1380035,1380035
mi_idx.movie_id,2,2525793,1380035,459925
mi_idx.info_type_id,99,113,1380035,5
mk.id,1,4523930,4523930,4523930
mk.movie_id,2,2525971,4523930,476794
mk.keyword_id,1,134170,4523930,134170
```

每张表的最小值、最大值、元组数目、不重复记录数目

Submission sample



```
Query ID,Predicted Cardinality
0,0
1,0
2,2
3,6
4,12
5,20
```

A naïve implementation

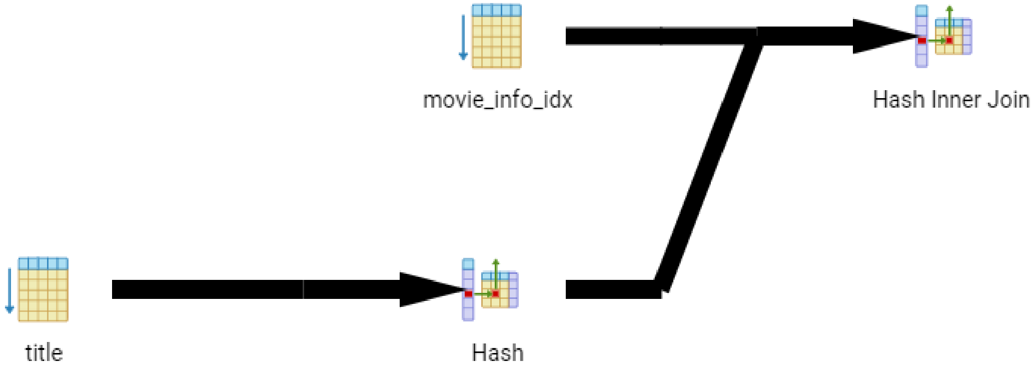
- 共有N个表，d个属性
- lb: lower bound   ub: upper bound
- 2*d维向量 $[lb_1,ub_1,lb_2,ub_2,\ldots\ldots.lb_d,ub_d]$
- One-hot encoding for join
- SELECT * FROM title t, movie_keyword mk WHERE t.production_year >2005 AND mk.keyword_id<1029 AND t.id=mk.movie_id

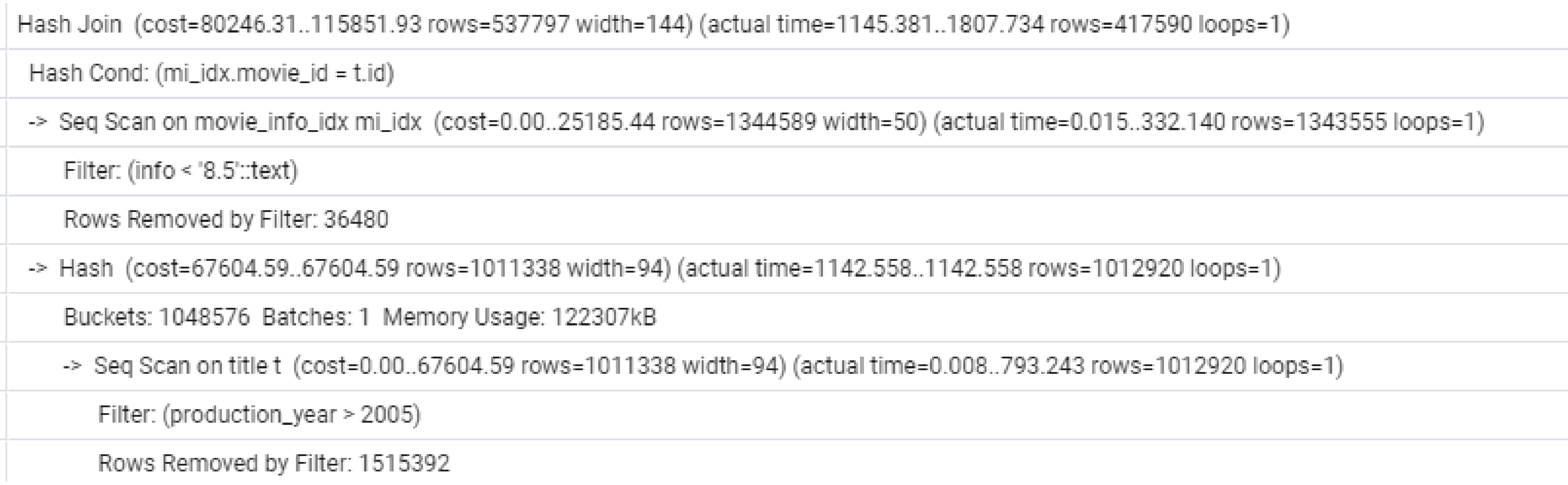
$$[0,\ 0,\ \underbrace{2005,\ 0}_{\text{t.production\_year}},\ldots\ldots,0,\ 0,\ \underbrace{0,\ 1029}_{\text{mk.keyword\_id}},\ 0,\ 0]$$

$$\underbrace{[0,\ 1,\ 0,\ \ldots\ldots,\ 0]}_{\text{t.id=mk.movie\_id}}$$

Query Plan (optimal)

```
SELECT * FROM
        title AS t,
        movie_info_idx AS mi_idx
WHERE
        t.production_year > 2005
        AND
        mi_idx.info < 8.5
        AND
        t.id = mi_idx.movie_id
```





```
Hash Join  (cost=80246.31..115851.93 rows=537797 width=144) (actual time=1145.381..1807.734 rows=417590 loops=1)
 Hash Cond: (mi_idx.movie_id = t.id)
 ->  Seq Scan on movie_info_idx mi_idx  (cost=0.00..25185.44 rows=1344589 width=50) (actual time=0.015..332.140 rows=1343555 loops=1)
       Filter: (info < '8.5'::text)
       Rows Removed by Filter: 36480
 ->  Hash  (cost=67604.59..67604.59 rows=1011338 width=94) (actual time=1142.558..1142.558 rows=1012920 loops=1)
       Buckets: 1048576  Batches: 1  Memory Usage: 122307kB
       ->  Seq Scan on title t  (cost=0.00..67604.59 rows=1011338 width=94) (actual time=0.008..793.243 rows=1012920 loops=1)
             Filter: (production_year > 2005)
             Rows Removed by Filter: 1515392
```

https://www.postgresql.org/docs/10/using-explain.html