Veritabanı Yönetim Sistemleri

Dr. Öğr. Üyesi Ahmet Arif AYDIN

Query Optimization

Sorgu Optimizasyonu

Özet

- Dosya Yapıları
  - Pages,
  - File Layer
  - Disk space manager
  - Buffer manager
- Index yapıları
  - Hash & Tree based indexing
- B+ Tree
  - search, insert , delete

Sorgu Optimizasyonu

Veritabanında kayıtlı olan veriler üzerinde sorgulama işlemini gerçekleştirmek için SQL dili kullanılır

Sorgular (query)

oluşmaktadır.

Sorgu Optimizasyonu

Veritabanında kayıtlı olan veriler üzerinde sorgulama işlemini gerçekleştirmek için
SQL dili kullanılır

Sorgular (query)
SQL komutları (Select, insert, create, alter, …..)
ve
çeşitli operatörler
(and, or, like, - , + , between, exists, …) 'den

oluşmaktadır.

Sorgu Optimizasyonu

SQL sorguları hesaplanırken
(işleme tabi tutulurken)
alternatif yollar ile
gerçekleştirilir.

Sorgu Optimizasyonu

SQL sorguları hesaplanırken
(işleme tabi tutulurken)
alternatif yollar ile
gerçekleştirilir.

Alternatif yollar
sorgu içerisinde kullanılan **farklı operatörlerin**
ve operatörlerin sağladığı işlemi gerçekleştiren
algoritmalarının farklı biçimde birleştirilmeleriyle
ortaya çıkmaktadır.

Sorgu Optimizasyonu

SQL sorguları hesaplanırken
(işleme tabi tutulurken)
alternatif yollar ile
gerçekleştirilir.

Alternatif yollar
sorgu içerisinde kullanılan **farklı operatörlerin**
ve operatörlerin sağladığı işlemi gerçekleştiren
algoritmalarının farklı biçimde birleştirilmeleriyle
ortaya çıkmaktadır.

Sorgunun hesaplanması işleminde
en iyi (zaman, kaynak kullanımı) **hesaplama planını**
bulma sürecine sorgu optimizasyonu denir.

Sorgu Değerlendirme (Query Evaluation)

Sorgu İşleme alma – değerlendirme aşamaları (Query Evaluation)

1. Sorgu **İlişkisel Cebir** formuna dönüştürülür.
2. Kullanılan operatörlere göre işlem ağacı oluşturulur.
3. Her bir işlem için kullanılacak algoritmalar belirlenir.

Sorgu Değerlendirme (Query Evaluation)

- Her bir operator için birden fazla algoritma bulunmaktadır.
- Gerçekleştirilecek işlemde hangi algoritmanın kullanılacağı ve daha iyi sonuç vereceğini aşağıdaki faktörler belirler:

Sorgu Değerlendirme (Query Evaluation)

- Her bir operator için birden fazla algoritma bulunmaktadır.
- Gerçekleştirilecek işlemde hangi algoritmanın kullanılacağı ve daha iyi sonuç vereceğini aşağıdaki faktörler belirler:
  - Tablolar da tutulan veri miktarı (size)
  - Varolan indexler ve sıralama (ordering)
  - Kullanılabilen bellek miktarı (RAM)

Sorgu Değerlendirme (Query Evaluation)

Sorgu işlemlerinde kullanılan algoritmalar aşagıdaki tekniklerle değerlendirme işlemlerini gerçekleştirirler

- **İndexleme (indexing) :** WHERE ifadesinden sonra tanımlananan şartlar ile daha az veri üzerinde işlem gerçekleştirilir (selections, joins)

Sorgu Değerlendirme (Query Evaluation)

Sorgu işlemlerinde kullanılan algoritmalar aşagıdaki tekniklerle değerlendirme işlemlerini gerçekleştirirler

- **İndexleme (indexing) :** WHERE ifadesinden sonra tanımlananan şartlar ile daha az veri üzerinde işlem gerçekleştirilir (selections, joins)

- **Döngü (iteration) :** Tablolarda bulunan değerler veya sadece index tabloları taranır.

- **Bölümlendirme (partitioning):** Sıralama (sorting) veya hash (hashing) ile üzerinde işlem yapılcak tuple sayısı değiştirilebilir.

Bağlantı Yolu (Access Path)

Tablolarda bulunan satırları (tuple) elde etme metoduna **access path** denir ve iki biçimde gerceklesir.

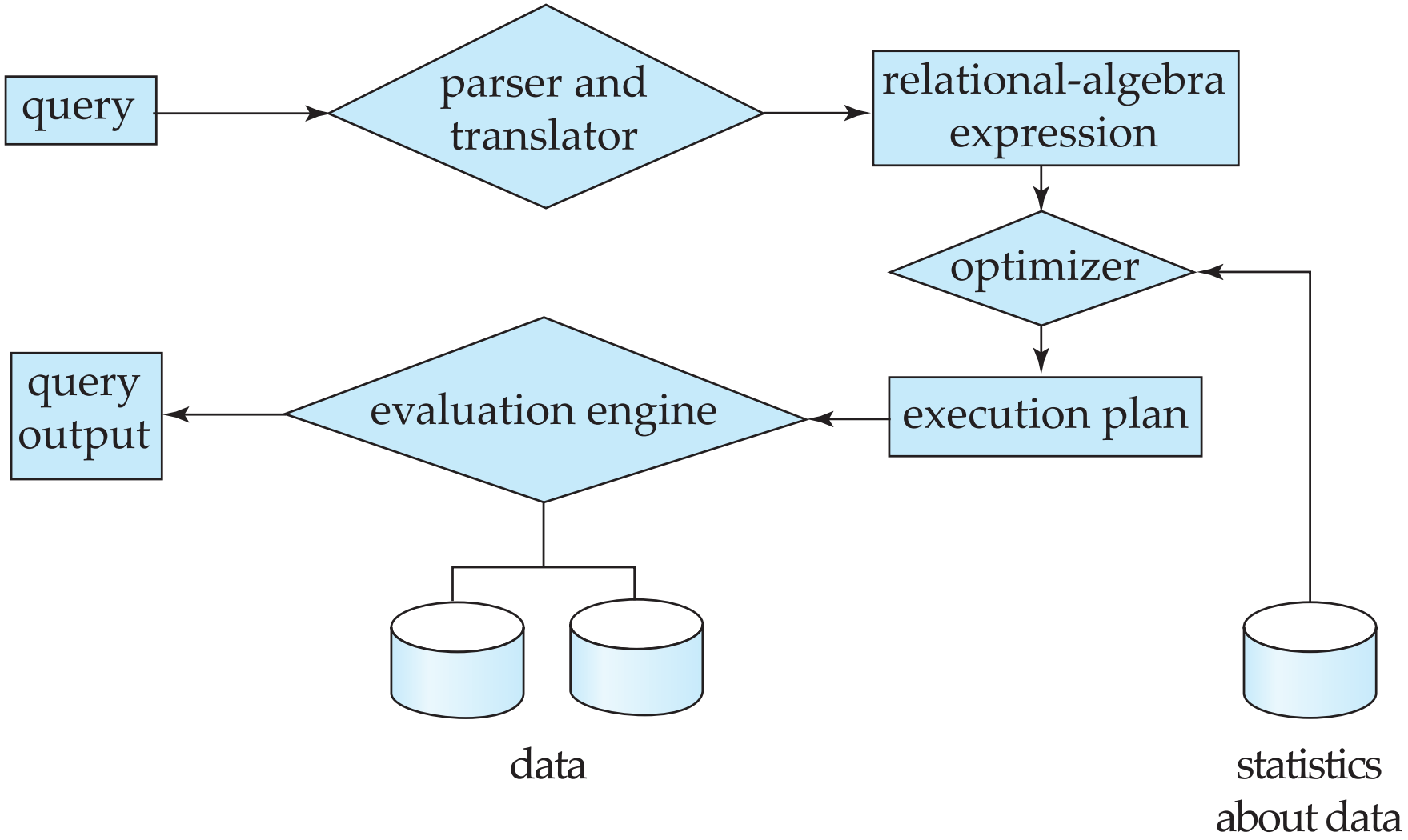
1. Dosya tarama (file scan)
2. Index + eşleştirme işlemi (index + matching selection)

Bağlantı Yolu (Access Path)

- Tree index
  - Arama işleminde **attribute** (<, > , = ,) **value** tanımlamasına göre eşleştirme yapar.
  - (a,b,e) arama anahtarı içerisinde a ile başlayan sorgular gerçekleştirilir
  - (a) veya (a,b) aranır fakat (b,e) aranmaz..

- Hash index
  - Arama indexinde **attribute= value** prensibine göre eşleştirme yapar.

Sorgunun Gerçekleştirilmesi



Veritabanı İlişkisel İşlem Algoritmaları

- Seçim (**Selection-** σ) : istenilen alan index değilse tablo taranır.
- izdüşüm-Yansıtma (**Projection-** π) : istenmeyen sütunları gizler
- **join** ($\bowtie$): birden fazla tabloyu belirlenen şartlara göre birleştirir. Maliyetli ve çok kullanılan bir işlemdir.

Veritabanı İlişkisel İşlem Algoritmaları (Selection)

Person

Name	Age	Weight
Harry	34	80
Sally	28	64
George	29	70
Helena	54	54
Peter	34	80

$\sigma_{\mathrm{Age} \geq 34}(\mathrm{Person})$

Name	Age	Weight
Harry	34	80
Helena	54	54
Peter	34	80

$\sigma_{\mathrm{Age} = \mathrm{Weight}}(\mathrm{Person})$

Name	Age	Weight
Helena	54	54

$\sigma_{\mathrm{Age} \geq 30 \ \wedge \ \mathrm{Weight} \leq 60}(\mathrm{Person})$

Name	Age	Weight
Helena	54	54

https://en.wikipedia.org/wiki/Selection_(relational_algebra)

Veritabanı İlişkisel İşlem Algoritmaları (Projection)

Person

Name	Age	Weight
Harry	34	180
Sally	28	164
George	28	170
Helena	54	154
Peter	34	180

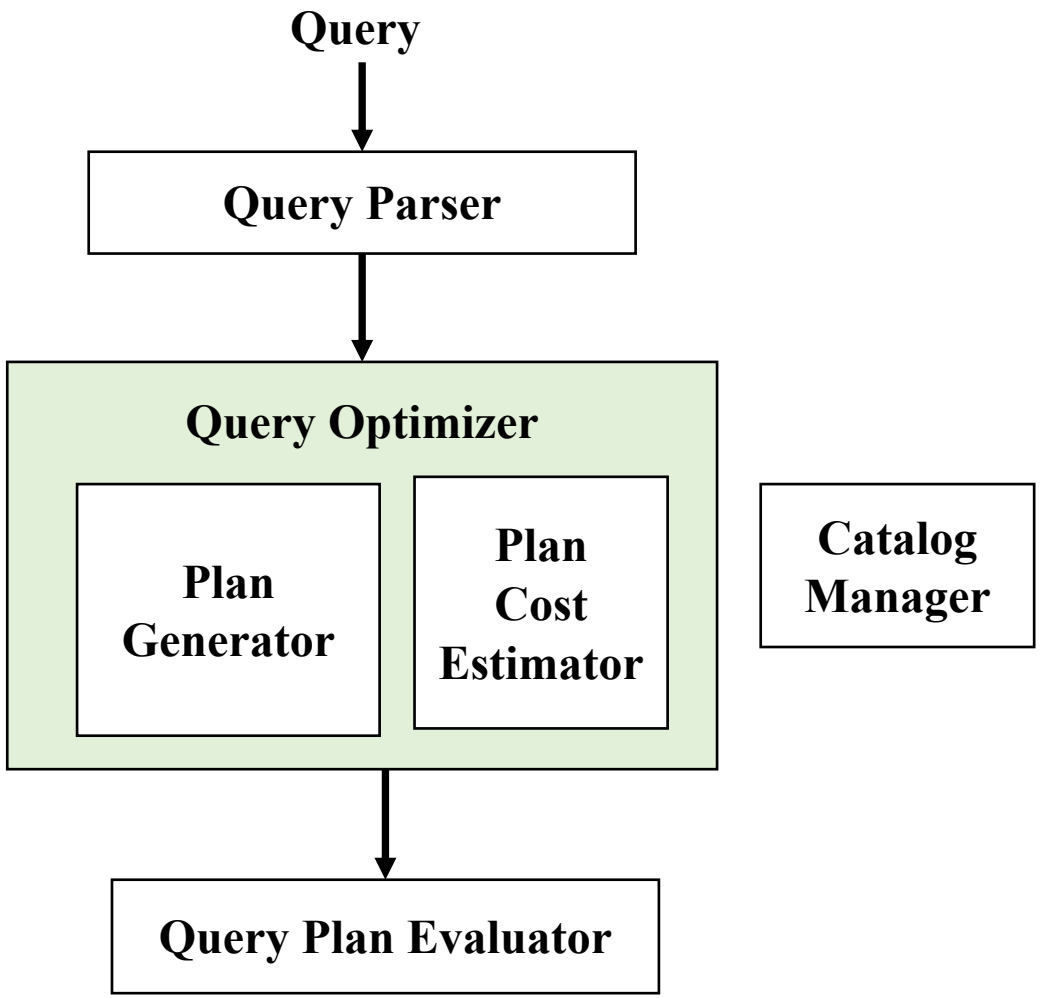
$\Pi_{\text{Age,Weight}}(\text{Person})$

Age	Weight
34	180
28	164
28	170
54	154

https://en.wikipedia.org/wiki/Projection_(relational_algebra)

Sorgu Optimizatörü (Query Optimizer)

Query Optimizer sayesinde bir sorgunun **gerçekleştirme planları ve her bir planın maliyet hesapları** bulunur ve sorgunun cevaplanması için en uygun plan uygulanır.



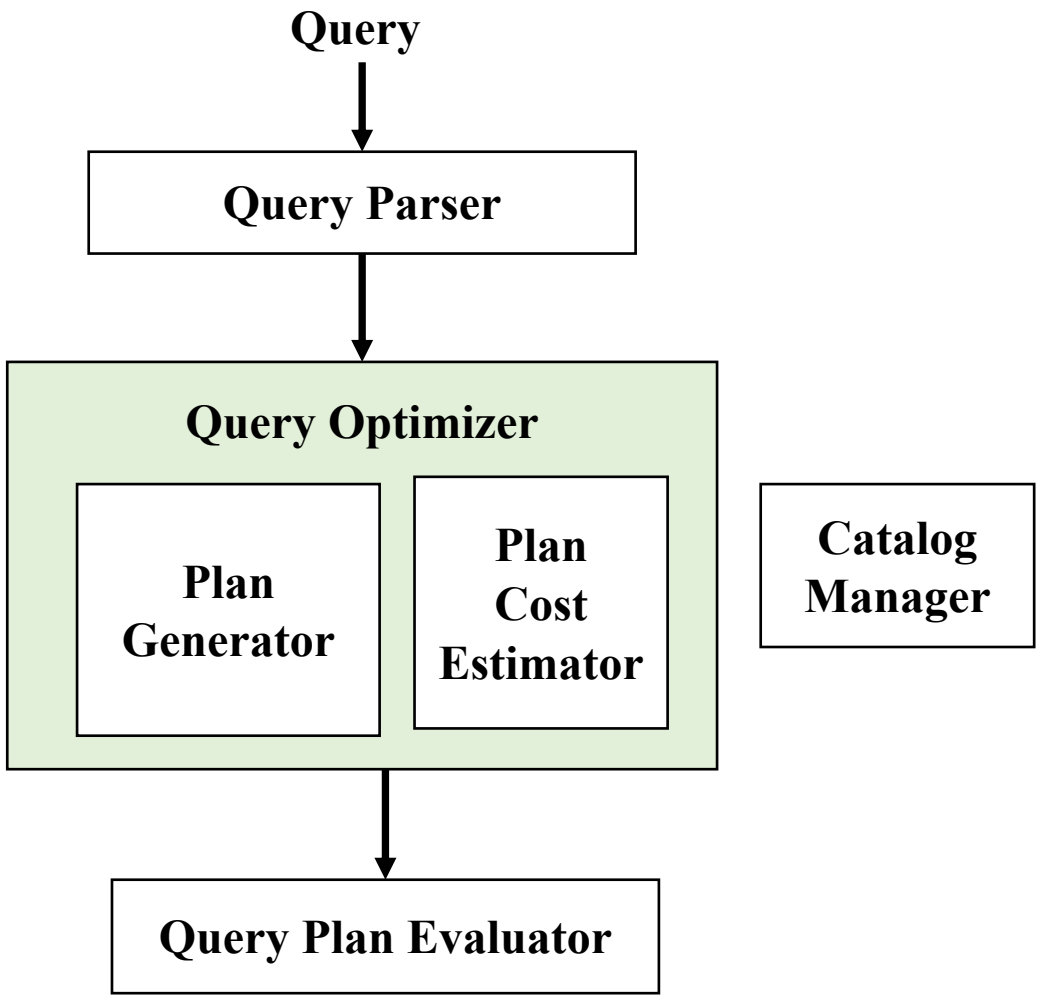
Sorgu Optimizatörü (Query Optimizer)

Query Optimizer sayesinde bir sorgunun **gerçekleştirme planları ve her bir planın maliyet hesapları** bulunur ve sorgunun cevaplanması için en uygun plan uygulanır.

a query is essentially treated as a

σ (selection), projection (π) and join (⋈)

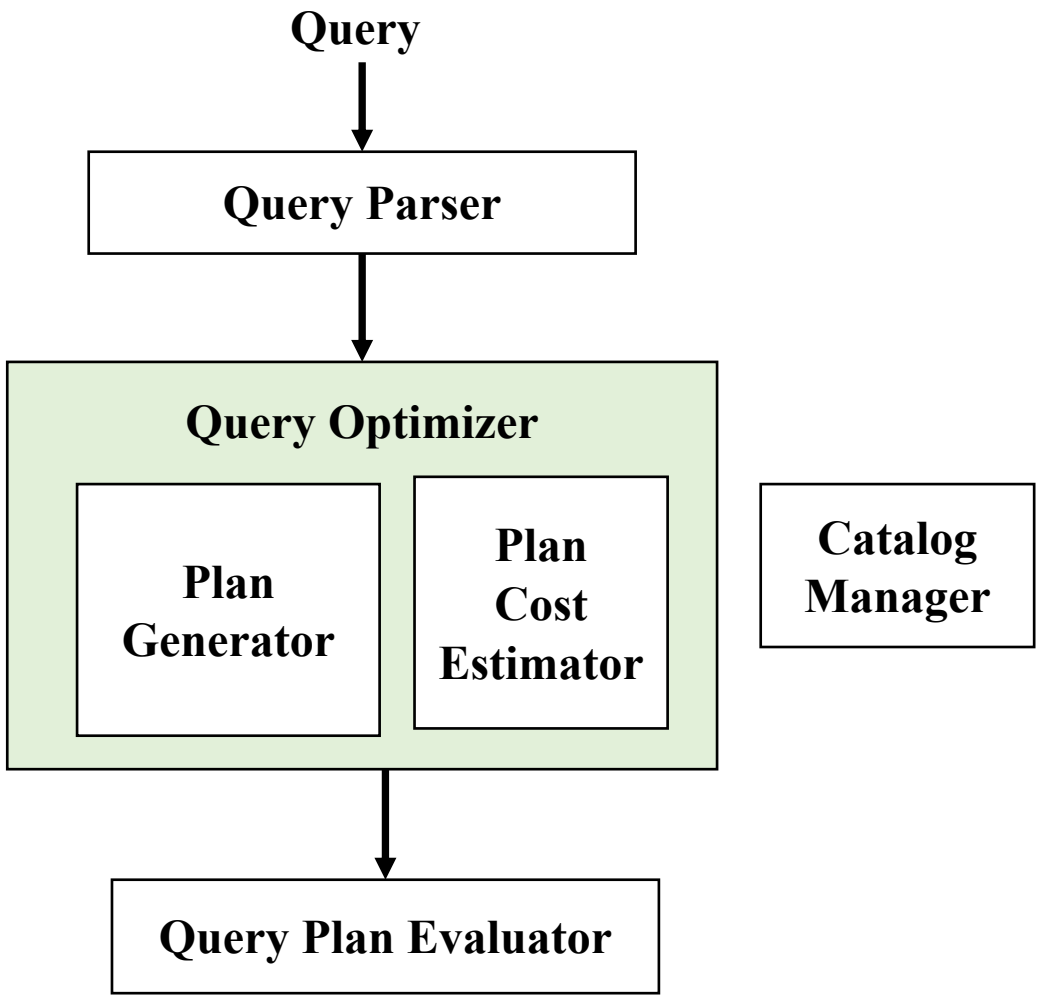
σ algebra expression



Sorgu Optimizatörü (Query Optimizer)

$A \bowtie B \bowtie C \bowtie D$

Kaç farklı biçimde join işlemi gerçekleştirilir?

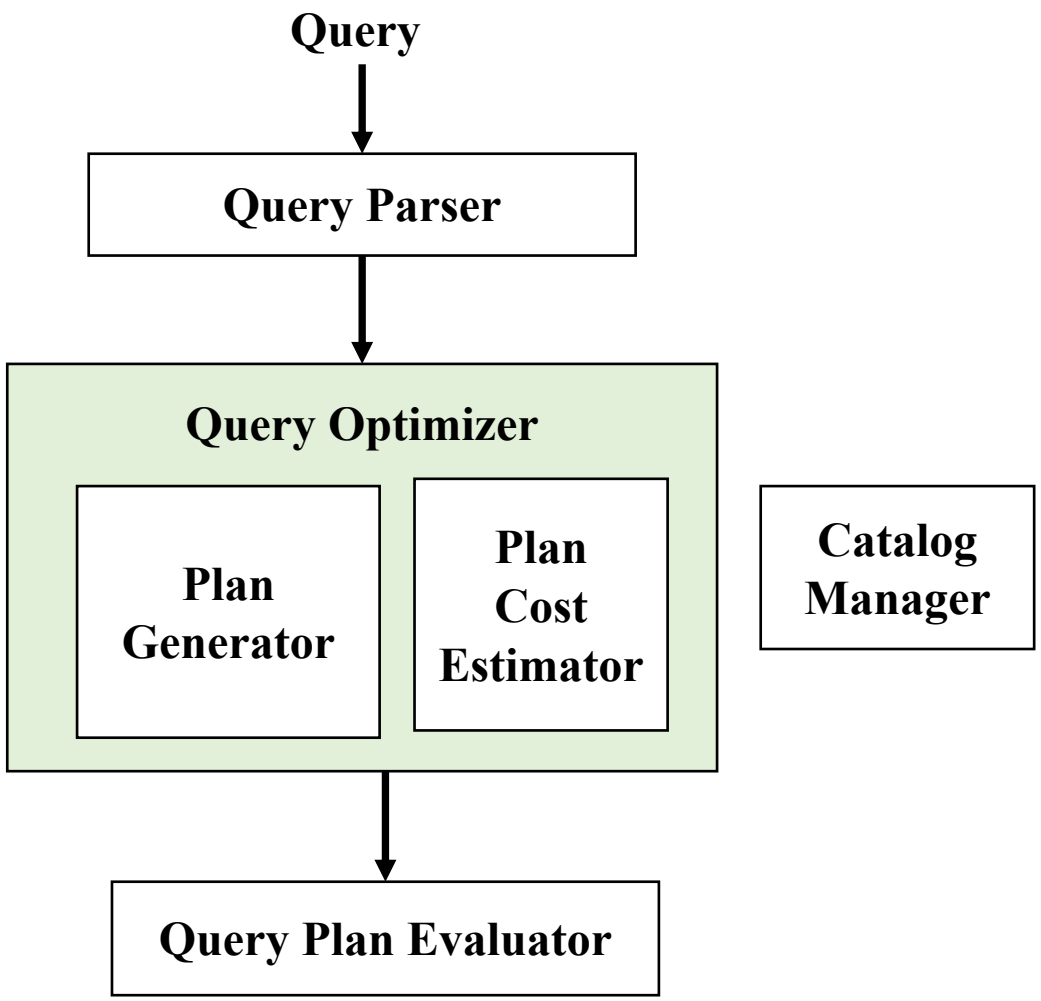


Sorgu Optimizatörü (Query Optimizer)

$A \bowtie B \bowtie C \bowtie D$

Kaç farklı biçimde join işlemi gerçekleştirilir?

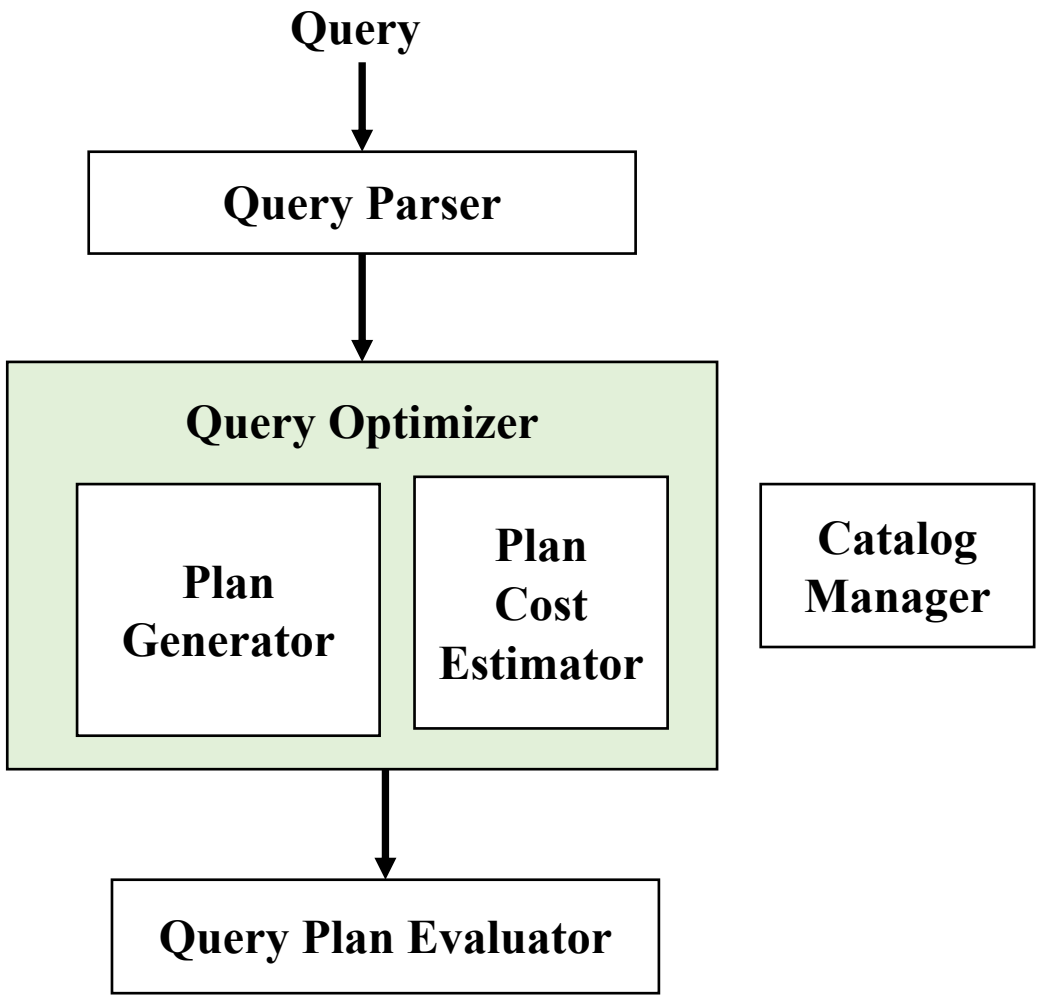
$4! = 24$



Sorgu Optimizatörü (Query Optimizer)

$A \bowtie B \bowtie C \bowtie D$

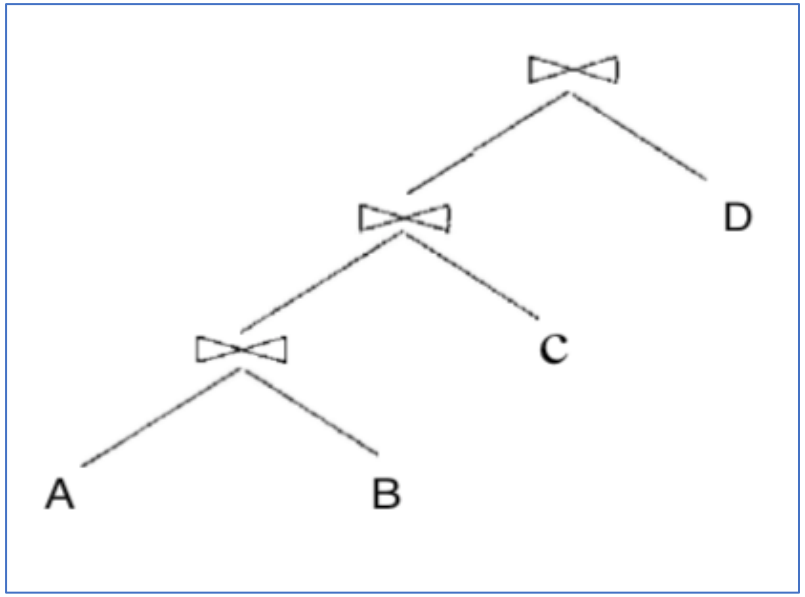
join işlemi nasıl gerçekleştirilir?

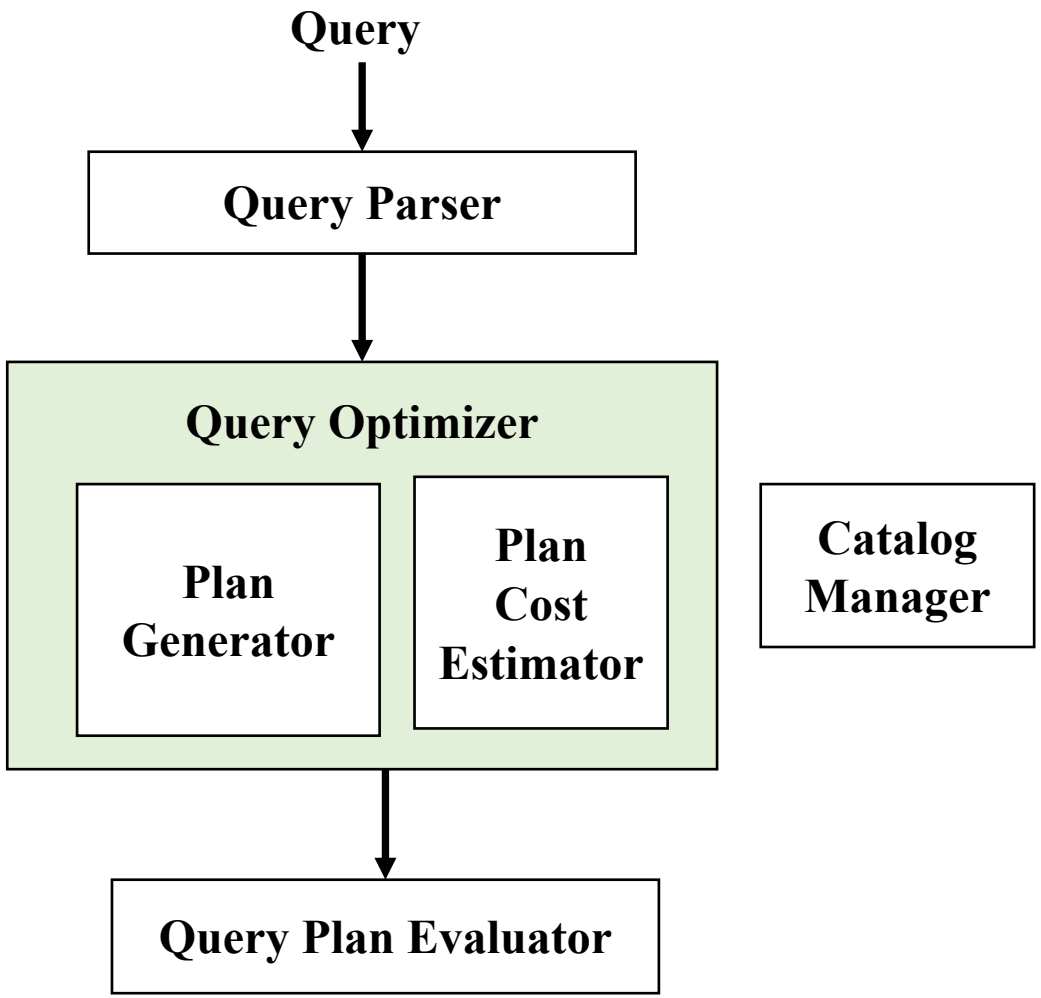


Sorgu Optimizatörü (Query Optimizer)

$(((A \bowtie B) \bowtie C) \bowtie D)$

join işlemi nasıl gerçekleştirilir?





Örnek Tablolar

Sailor

sid	sname	reyting	
22	Dustin	7	45.0
29	Brutus	1	33.0
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35.0
64	Horatio	7	35.0
71	Zorba	10	16.0
74	Horatio	9	35.0
85	Art	3	25.5
95	Bob	3	63.5

Boat

bid	bname	color
101	Interlake	blue
102	Interlake	red
103	Clipper	green
104	Marine	red

Reserves

sid	bid	day
22	101	10/10/98
22	102	10/10/98
22	103	10/8/98
22	104	10/7/98
31	102	11/10/98
31	103	11/6/98
31	104	11/12/98
64	101	9/5/98
64	102	9/8/98
74	103	9/8/98

s: sailor

b: boat

Sorgu Değerlendirilmesi (Query Evaluation)

1

```
SELECT  S.sname
FROM    Reserves R, Sailors S
WHERE   R.sid = S.sid
        AND R.bid = 100 AND S.rating > 5
```

Sorgu Değerlendirilmesi (Query Evaluation)

1

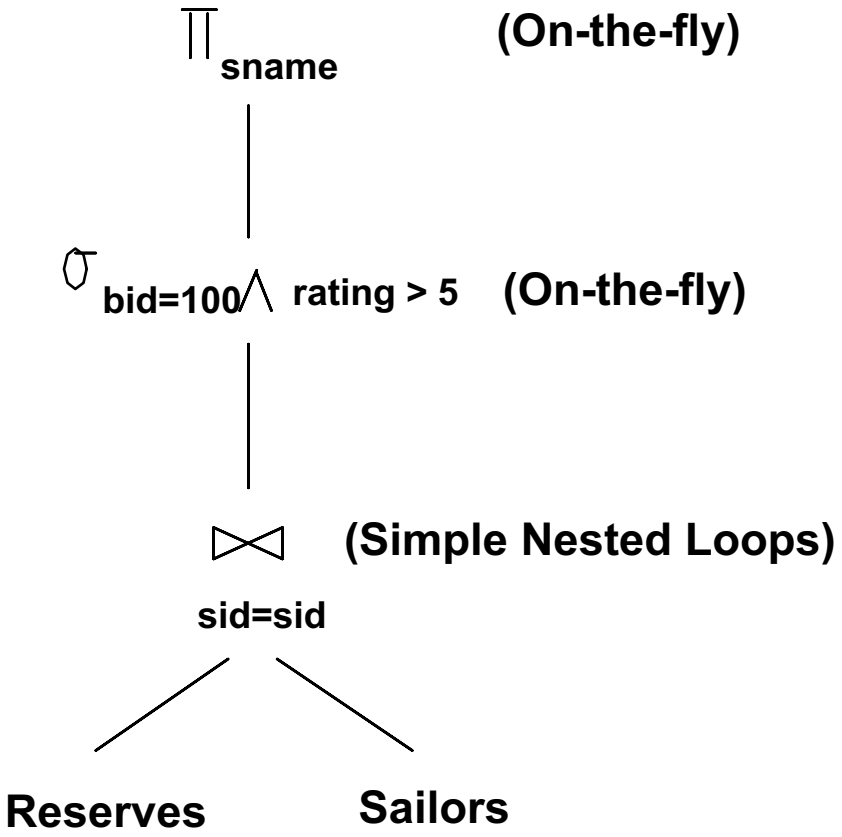
```
SELECT  S.sname
FROM    Reserves R, Sailors S
WHERE   R.sid = S.sid
        AND R.bid = 100 AND S.rating > 5
```

2

$$\pi_{sname}\left(\sigma_{bid=100 \wedge rating>5}\left(Reserves \bowtie_{sid=sid} Sailors\right)\right)$$

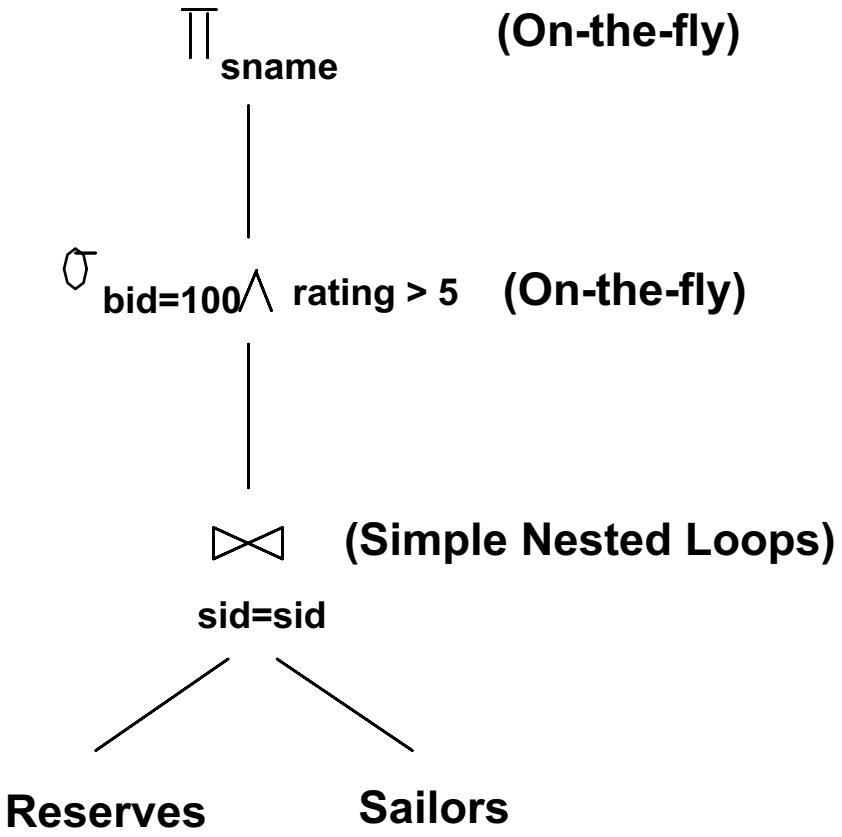
Sorgu Değerlendirilmesi (Query Evaluation)

3-a

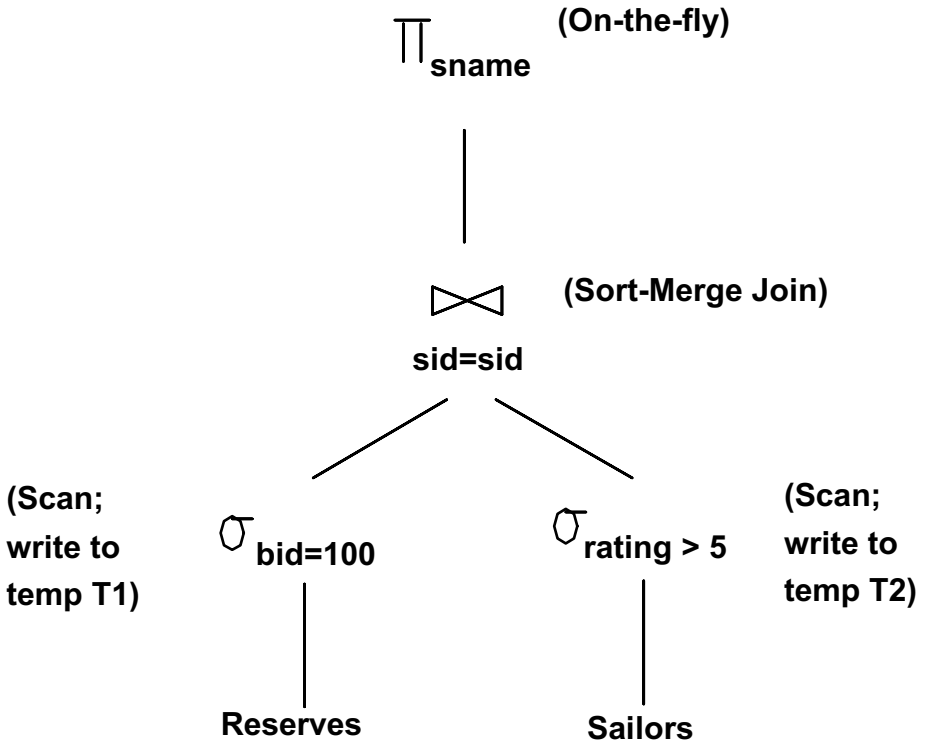


Sorgu Değerlendirilmesi (Query Evaluation)

3-a

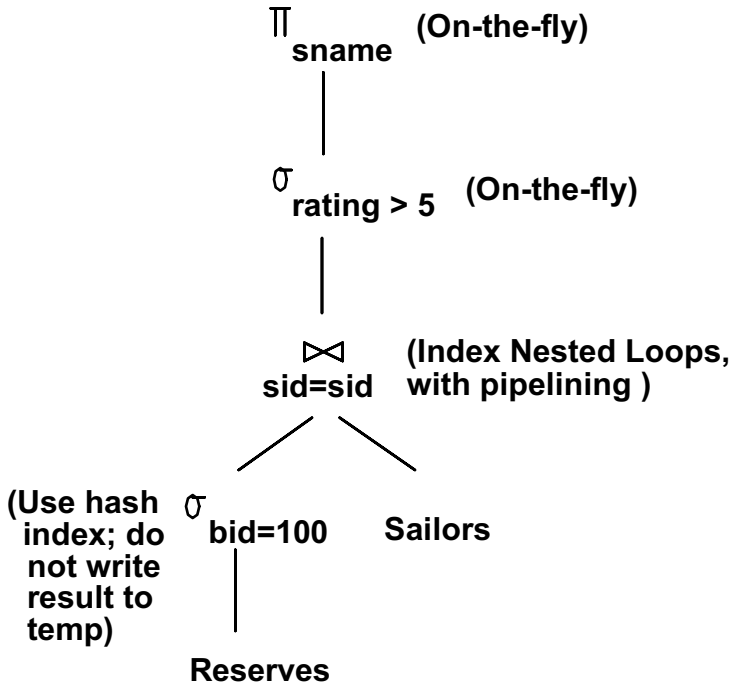


3-b

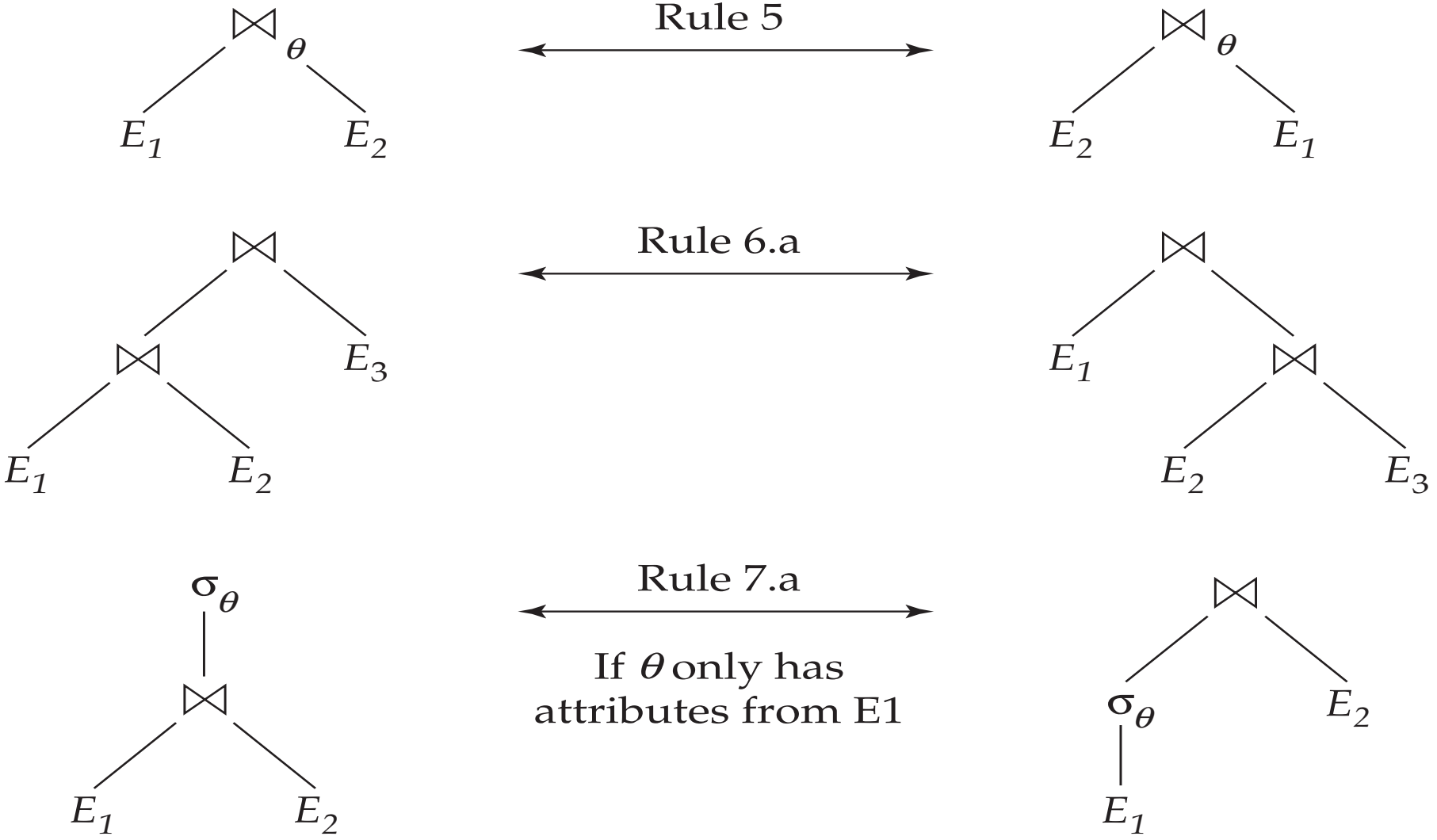


Sorgu Değerlendirilmesi (Query Evaluation)

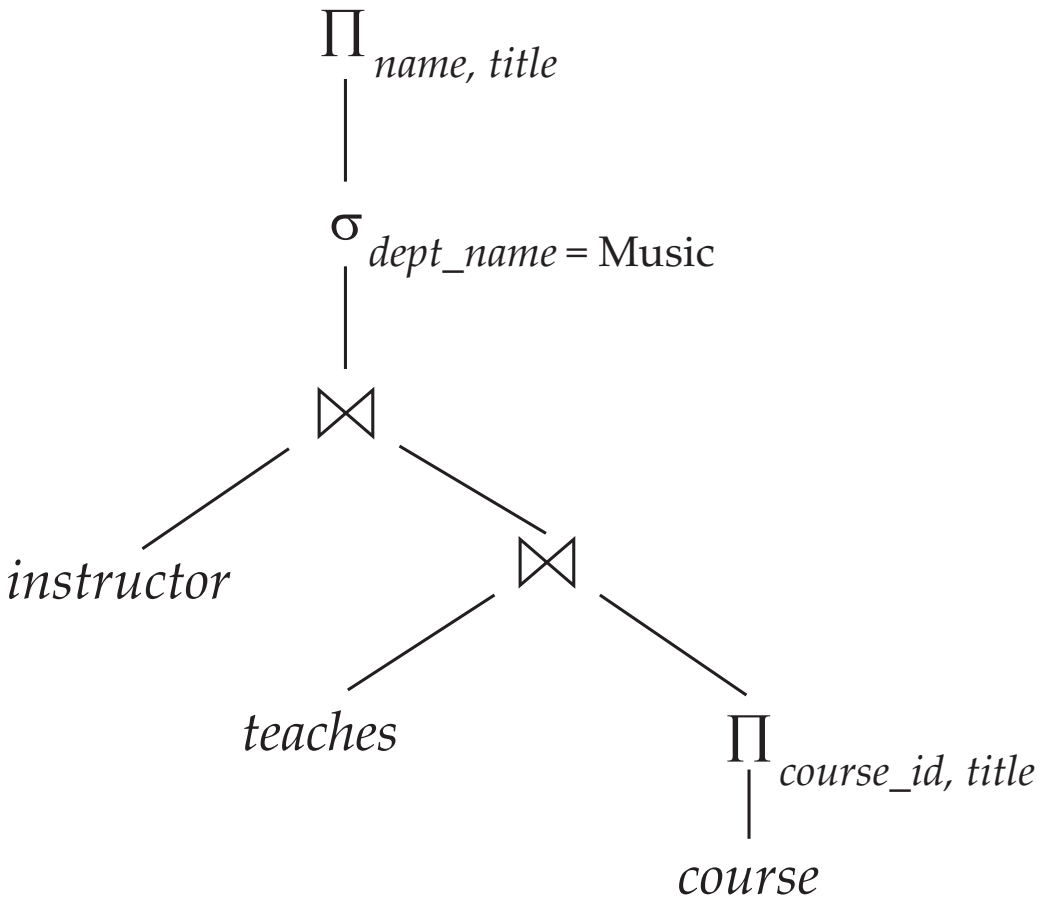
3-c



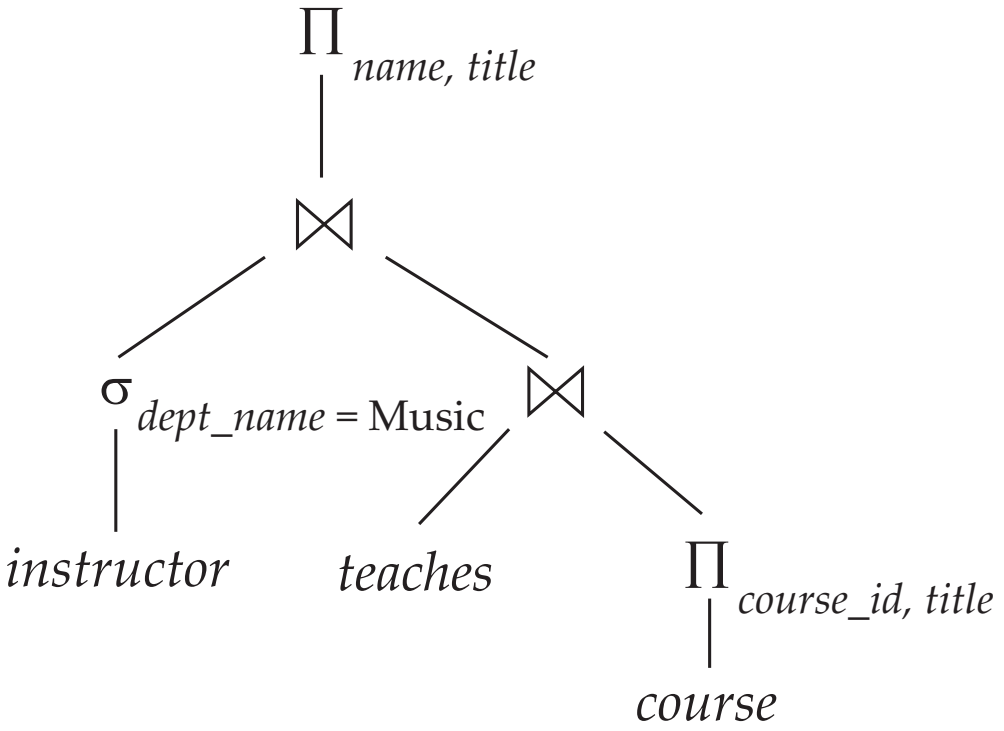
Sorgu Değerlendirilmesi (Query Evaluation): Alternatif Planlar



Örnek 2 : Sorgu Değerlendirilmesi (Query Evaluation): Alternatif Planlar

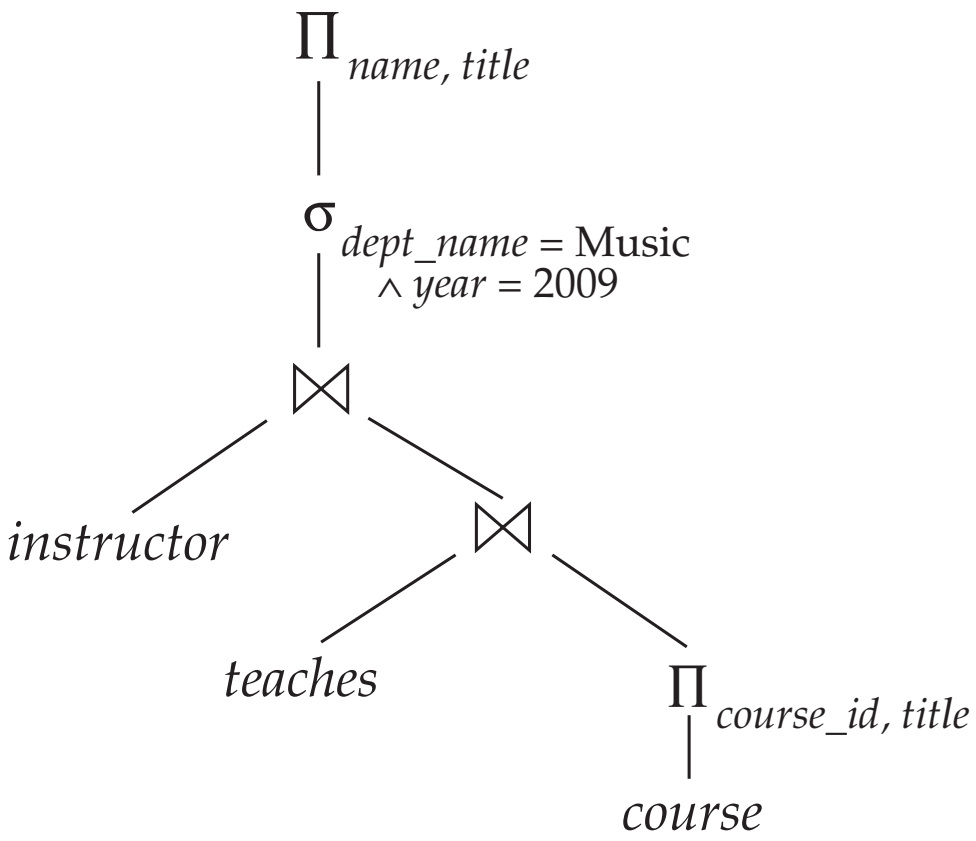


(a) Initial expression tree

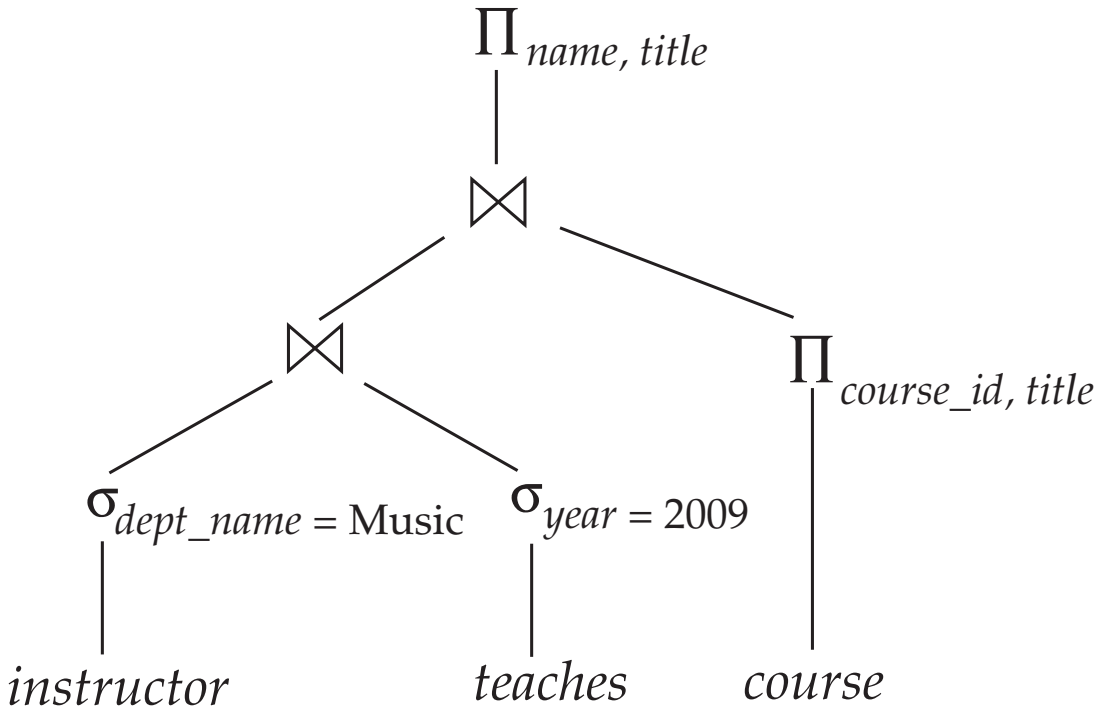


(b) Transformed expression tree

Örnek 4 : Sorgu Değerlendirilmesi (Query Evaluation): Alternatif Planlar



(a) Initial expression tree



(b) Tree after multiple transformations

PostgreSQL Sorgu Planı (Query Plan)


```
vtys17=# explain analyze select ögrenci.ögrencino, ögrenci.ögrenciadı,ögrenci.yaş from ögrenci, kayıt where ögrenci.ögrencino
= kayıt.ögrencino and kayıt.derskodu = ( Select derskodu from dersler where dersadı='Veritabanı Yönetim Sistemleri') order by
yaş desc ;
                                                    QUERY PLAN
-------------------------------------------------------------------------------------------------------------------
 Sort  (cost=58.28..58.29 rows=4 width=40) (actual time=0.100..0.100 rows=2 loops=1)
   Sort Key: "ögrenci"."yaş" DESC
   Sort Method: quicksort  Memory: 25kB
   InitPlan 1 (returns $0)
     ->  Seq Scan on dersler  (cost=0.00..18.12 rows=3 width=32) (actual time=0.010..0.011 rows=1 loops=1)
           Filter: (("dersadı")::text = 'Veritabanı Yönetim Sistemleri'::text)
           Rows Removed by Filter: 6
   ->  Hash Join  (cost=20.18..40.11 rows=4 width=40) (actual time=0.057..0.061 rows=2 loops=1)
         Hash Cond: ("ögrenci"."ögrencino" = "kayıt"."ögrencino")
         ->  Seq Scan on "ögrenci"  (cost=0.00..17.20 rows=720 width=40) (actual time=0.013..0.014 rows=8 loops=1)
         ->  Hash  (cost=20.12..20.12 rows=4 width=4) (actual time=0.033..0.033 rows=2 loops=1)
               Buckets: 1024  Batches: 1  Memory Usage: 9kB
               ->  Seq Scan on "kayıt"  (cost=0.00..20.12 rows=4 width=4) (actual time=0.025..0.027 rows=2 loops=1)
                     Filter: ((derskodu)::text = ($0)::text)
                     Rows Removed by Filter: 10
 Planning time: 0.880 ms
 Execution time: 0.192 ms
(17 rows)

vtys17=#
```

http://postgresguide.com/performance/explain.html

https://thoughtbot.com/blog/reading-an-explain-analyze-query-plan

PostgreSQL Sorgu Planı (Query Plan)


```
vtys17=# explain analyze select ögrenci.ögrencino, ögrenci.ögrenciadı,ögrenci.yaş from ögrenci, kayıt where ögrenci.ögrencino
= kayıt.ögrencino and kayıt.derskodu = ( Select derskodu from dersler where dersadı='Veritabanı Yönetim Sistemleri') order by
yaş desc ;
                                                                    QUERY PLAN
-----------------------------------------------------------------------------------------------------------------------------------
 Sort  (cost=58.28..58.29 rows=4 width=40) (actual time=0.100..0.100 rows=2 loops=1)
   Sort Key: "ögrenci"."yaş" DESC
   Sort Method: quicksort  Memory: 25kB
   InitPlan 1 (returns $0)
     ->  Seq Scan on dersler  (cost=0.00..18.12 rows=3 width=32) (actual time=0.010..0.011 rows=1 loops=1)
           Filter: (("dersadı")::text = 'Veritabanı Yönetim Sistemleri'::text)
           Rows Removed by Filter: 6
   ->  Hash Join  (cost=20.18..40.11 rows=4 width=40) (actual time=0.057..0.061 rows=2 loops=1)
         Hash Cond: ("ögrenci"."ögrencino" = "kayıt"."ögrencino")
         ->  Seq Scan on "ögrenci"  (cost=0.00..17.20 rows=720 width=40) (actual time=0.013..0.014 rows=8 loops=1)
         ->  Hash  (cost=20.12..20.12 rows=4 width=4) (actual time=0.033..0.033 rows=2 loops=1)
               Buckets: 1024  Batches: 1  Memory Usage: 9kB
               ->  Seq Scan on "kayıt"  (cost=0.00..20.12 rows=4 width=4) (actual time=0.025..0.027 rows=2 loops=1)
                     Filter: ((derskodu)::text = ($0)::text)
                     Rows Removed by Filter: 10
 Planning time: 0.880 ms
 Execution time: 0.192 ms
(17 rows)

vtys17=#
```

http://postgresguide.com/performance/explain.html

https://thoughtbot.com/blog/reading-an-explain-analyze-query-plan

Sistem Kataloğu (System Catalog)

- VTYS'lerinin sağladığı ortamda oluşturulan **tablolar ve index yapısı** veri (data) olarak tanımlanır.

- VTYS'de oluşturulan **her tablo** ve **index yapısı** bilgisini içeren **tanımlayıcı bilgilerden oluşan tablo koleksiyonu** bulunmaktadır.

Sistem Kataloğu (System Catalog)

- VTYS'lerinin sağladığı ortamda oluşturulan **tablolar ve index yapısı** veri (data) olarak tanımlanır.

- VTYS'de oluşturulan **her tablo** ve **index yapısı** bilgisini içeren **tanımlayıcı bilgilerden oluşan tablo koleksiyonu** bulunmaktadır.

Tablolar ve index yapısının kaydedilip yönetimin gerçekleştirildiği yapı

system catalog, catalog tables, data dictionary, catalog

olarak da isimlendirilmektedir.

Sistem Kataloğu (System Catalog)

Sistem kataloğunda aşagıdaki bilgiler bulunur

- Tablolar (tablo ismi , kolon isimleri tipleri, index isimleri, PK, FK)
- indexler (index adı, yapısı (B+) , arama anahtarları)
- View (isim ve tanımlama)
- Cardinality (tablodaki tuple sayısı)
- Size (sayfa sayısı)
- Nkeys (index sayısı)
- Ağaç yapısının yüksekliği
- index aralığı (range) min, max

Sistem Kataloğu (System Catalog)

Sistem kataloğunda aşagıdaki bilgiler bulunur

- Tablolar (tablo ismi , kolon isimleri tipleri, index isimleri, PK, FK)
- indexler (index adı, yapısı (B+) , arama anahtarları)
- View (isim ve tanımlama)
- Cardinality (tablodaki tuple sayısı)
- Size (sayfa sayısı)
- Nkeys (index sayısı)
- Ağaç yapısının yüksekliği
- index aralığı (range) min, max

Cataloglar periyodik olarak güncelleniyor (tabloların her işlemde değiştirilmesi ve güncellenmesi maliyetli)

Sistem Kataloğu (System Catalog)


```
SELECT  * FROM pg_catalog.pg_tables;
```

Data Output | Explain | Messages | Query History

	schemaname name	tablename name	tableowner name	tablespace name	hasindexes boolean	hasrules boolean	hastriggers boolean	rowsecurity boolean
1	pg_catalog	pg_statistic	ahmetaydin	[null]	true	false	false	false
2	pg_catalog	pg_type	ahmetaydin	[null]	true	false	false	false
3	public	ögrenci	ahmetaydin	[null]	true	false	true	false
4	pg_catalog	pg_policy	ahmetaydin	[null]	true	false	false	false
5	pg_catalog	pg_authid	ahmetaydin	pg_global	true	false	false	false
6	public	kayıt	ahmetaydin	[null]	false	false	true	false
7	public	dersler	ahmetaydin	[null]	true	false	true	false
8	public	bölüm	ahmetaydin	[null]	true	false	true	false
9	pg_catalog	pg_user_ma...	ahmetaydin	[null]	true	false	false	false
10	pg_catalog	pg_subscript...	ahmetaydin	pg_global	true	false	false	false
11	pg_catalog	pg_attribute	ahmetaydin	[null]	true	false	false	false
12	pg_catalog	pg_proc	ahmetaydin	[null]	true	false	false	false
13	pg_catalog	pg_class	ahmetaydin	[null]	true	false	false	false
14	pg_catalog	pg_attrdef	ahmetaydin	[null]	true	false	false	false
15	pg_catalog	pg_constraint	ahmetaydin	[null]	true	false	false	false
16	pg_catalog	pg_inherits	ahmetaydin	[null]	true	false	false	false
17	pg_catalog	pg_index	ahmetaydin	[null]	true	false	false	false
18	pg_catalog	pg_operator	ahmetaydin	[null]	true	false	false	false
19	pg_catalog	pg_opfamily	ahmetaydin	[null]	true	false	false	false
20	pg_catalog	pg_opclass	ahmetaydin	[null]	true	false	false	false

Sistem Kataloğu (System Catalog)

```
SELECT * FROM pg_catalog.pg_index;
```

Data Output | Explain | Messages | Query History

	indexrelid oid	indrelid oid	indnatts smallint	indisunique boolean	indisprimary boolean	indisexclusion boolean	indimmediate boolean	indisclustered boolean	indisvalid boolean	indcheckxmin boolean	indisready boolean
1	2831	2830	2	true	true	false	true	false	true	false	true
2	2833	2832	2	true	true	false	true	false	true	false	true
3	2835	2834	2	true	true	false	true	false	true	false	true
4	2837	2836	2	true	true	false	true	false	true	false	true
5	2839	2838	2	true	true	false	true	false	true	false	true
6	3599	3598	2	true	true	false	true	false	true	false	true
7	2841	2840	2	true	true	false	true	false	true	false	true
8	3440	3439	2	true	true	false	true	false	true	false	true
9	2337	2336	2	true	true	false	true	false	true	false	true
10	2847	2846	2	true	true	false	true	false	true	false	true