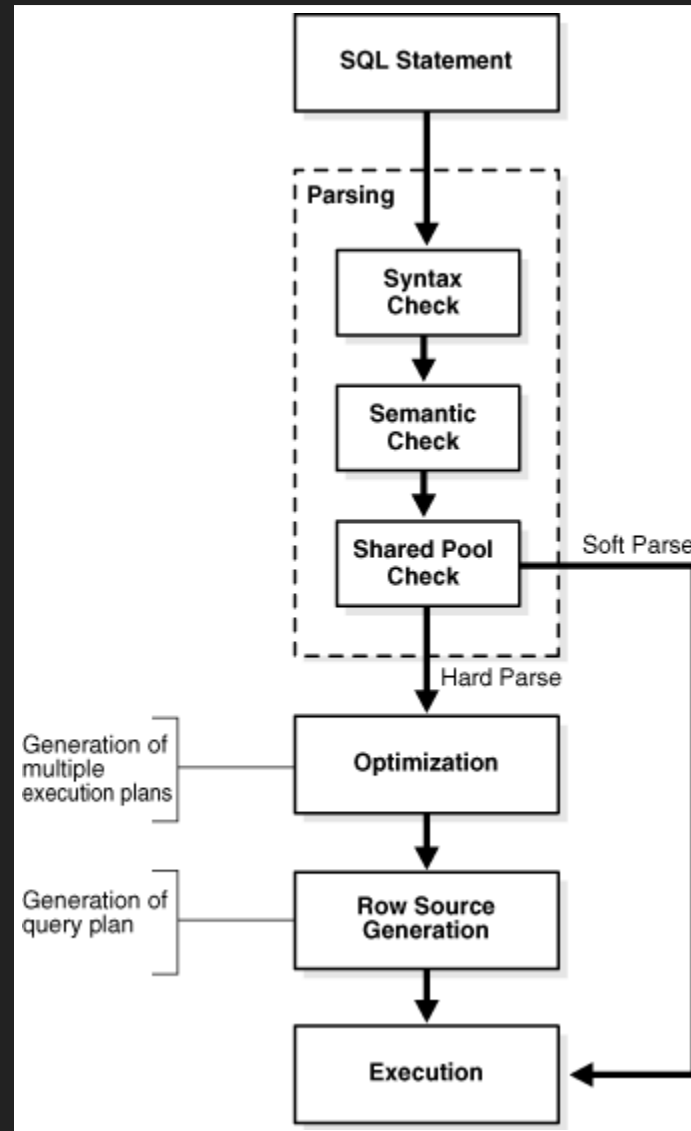


# QUERY OPTIMIZER

Created by [Jongwon](#)

# SQL PROCESSING



# SQL PARSING

구문분석이라고 하며  
다른 루틴을 처리할 수 있는 자료구조로  
SQL을 부분으로 분리하는 것을 포함한다.

- Syntax Check
- Semantic Check
- Shared Pool Check

# SYNTAX CHECK

다음 오류는 Syntax Check를 실패한 화면입니다.

```
12:03:09 HR> SELECT * FORM employees;  
SELECT * FORM employees  
          *  
ERROR at line 1:  
ORA-00923: FROM keyword not found where expected
```

SQL 문장 자체에서 구문을 나눌 수 있는지 체크합니다.

위 오류는 예상한 위치에 키워드를 찾을 수 없어서 발생한 것입니다.

# SEMANTIC CHECK

다음 오류는 Semantic Check를 실패한 화면입니다.

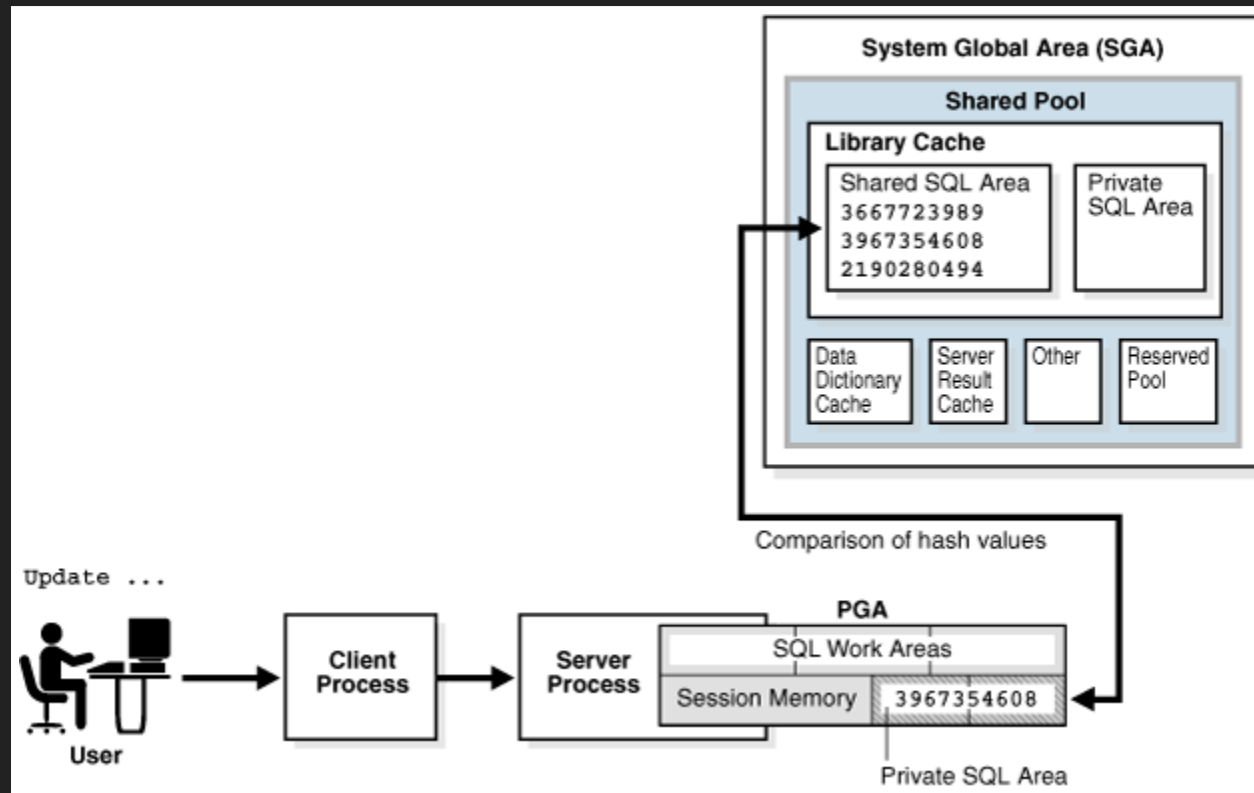
```
16:17:19 HR> SELECT * FROM 1;  
SELECT * FROM 1  
          *  
ERROR at line 1:  
ORA-00903: invalid table name
```

```
16:19:41 HR> SELECT * FROM ASDFASDF;  
SELECT * FROM ASDFASDF  
          *  
ERROR at line 1:  
ORA-00942: table or view does not exist
```

각 나누어진 구문에서의 의미를 체크합니다.

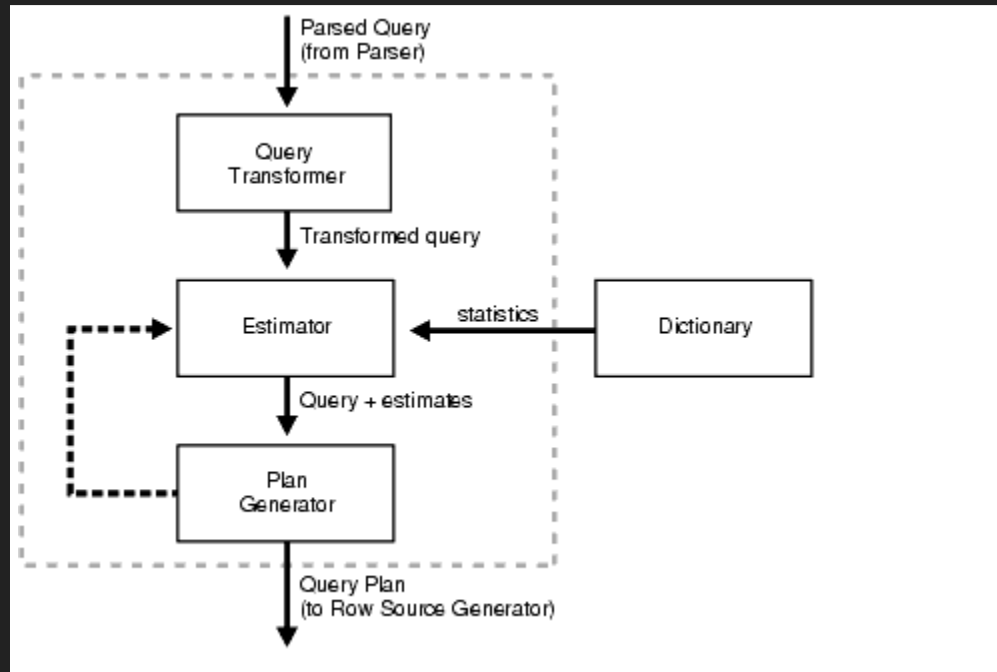
위 오류는 테이블 이름의 의미가 잘못되어(왼쪽),  
테이블을 찾을 수 없어(오른쪽) 발생한 오류입니다.

# SHARED POOL CHECK



SQL ID를 사용하여 메모리 주소를 확인하거나  
**Execution Plan**의 해시값을 확인하여  
리소스를 많이 사용하는 단계를 건너 뛸 수 있습니다.

# EXECUTION PLAN



Parsed 된 쿼리를 Execution Plan으로 출력하는 과정은 위와 같습니다.

# EXECUTION PLAN 이란?

SQL문을 실행하기 위해 필요한  
구체적인 과정을 보여주는 것



# Displaying the Execution plan

```
-- SH(Sales History) Schema
SELECT prod_category, AVG(amount_sold)
FROM sales s, products p
WHERE p.prod_id = s.prod_id
GROUP BY prod_category;
```

Execution Plan

-----

Plan hash value: 1197568639

-----

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	Pstart	Pstop
0	SELECT STATEMENT		5	255	551 (6)	00:00:07		
1	HASH GROUP BY		5	255	551 (6)	00:00:07		
* 2	HASH JOIN		72	3672	550 (6)	00:00:07		
3	VIEW	VW_GBC_5	72	2160	548 (6)	00:00:07		
4	HASH GROUP BY		72	648	548 (6)	00:00:07		
5	PARTITION RANGE ALL		918K	8075K	525 (2)	00:00:07	1	28
6	TABLE ACCESS FULL	SALES	918K	8075K	525 (2)	00:00:07	1	28
7	VIEW	index\$_join\$_002	72	1512	2 (0)	00:00:01		
* 8	HASH JOIN		72	1512	1 (0)	00:00:01		
9	INDEX FAST FULL SCAN	PRODUCTS_PK	72	1512	1 (0)	00:00:01		
10	INDEX FAST FULL SCAN	PRODUCTS_PROD_CAT_IX	72	1512	1 (0)	00:00:01		

-----

Predicate Information (identified by operation id):

-----

2 - access("P"."PROD\_ID"="ITEM\_1")

8 - access(ROWID=ROWID)

Statistics

-----

1252	recursive calls
0	db block gets
3552	consistent gets
1681	physical reads
0	redo size
851	bytes sent via SQL*Net to client
524	bytes received via SQL*Net from client
2	SQL*Net roundtrips to/from client
116	sorts (memory)
0	sorts (disk)
5	rows processed

# AUTOTRACE 켜기

오라클 설치시 기본적으로 설치 되어 있지만 Disable되어있다.

```
Enter user-name: hr
Enter password:

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

12:58:17 HR> show all
appinfo is OFF and set to "SQL*Plus"
arraysize 15
autocommit OFF
autoprint OFF
autorecovery OFF
autotrace OFF
blockterminator "." (hex 2e)
btitle OFF and is the first few characters of the next SELECT statement
cmdsep OFF
colsep " "
compatibility version NATIVE
concat "." (hex 2e)
copycommit 0
COPYTYPECHECK is ON
```

# AUTOTRACE 켜기

```
CONN / as sysdba
```

```
@?/sqlplus/admin/plustrce.sql
```

```
GRANT plustrace to HR
```

```
CONN hr
```

```
@?/rdbms/admin/utlxplan.sql
```

```
SET autotrace on
```

# EXPLAIN FOR 구문을 이용

```
20:07:00 HR> EXPLAIN PLAN FOR
20:07:08 2 SELECT last_name FROM employees;

Explained.

20:07:18 HR> @?/rdbms/admin/utlxpls.sql

PLAN_TABLE_OUTPUT
-----
Plan hash value: 2228653197

-----
| Id | Operation          | Name       | Rows | Bytes | Cost (%CPU)| Time     |
-----+-----+-----+-----+-----+-----+-----+
|  0 | SELECT STATEMENT    |            |    20 |    140 |      1   (0)| 00:00:01 |
|  1 |  INDEX FULL SCAN    | EMP_NAME_IX|    20 |    140 |      1   (0)| 00:00:01 |
-----+-----+-----+-----+-----+-----+

8 rows selected.

20:07:50 HR> █
```

ed를 눌러 edit화면으로 들어가면

```
select plan_table_output from table(dbms_xplan.display('plan_table',null,'serial'))
/
```

# EXECUTION PLAN 순서

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	Pstart	Pstop
0	SELECT STATEMENT				1140 (100)			
1	HASH GROUP BY		4	80	1140 (45)	00:00:14		
* 2	HASH JOIN		489K	9555K	792 (21)	00:00:10		
3	TABLE ACCESS FULL	PRODUCTS	767	8437	10 (0)	00:00:01		
4	PARTITION RANGE ALL		489K	4300K	741 (17)	00:00:09	1	16
5	TABLE ACCESS FULL	SALES	489K	4300K	741 (17)	00:00:09	1	16

Figure 1: Tabular shaped execution plan

위의 표 형식의 Execution plan은 아래와 같이 트리구조로 해석할 수 있습니다.

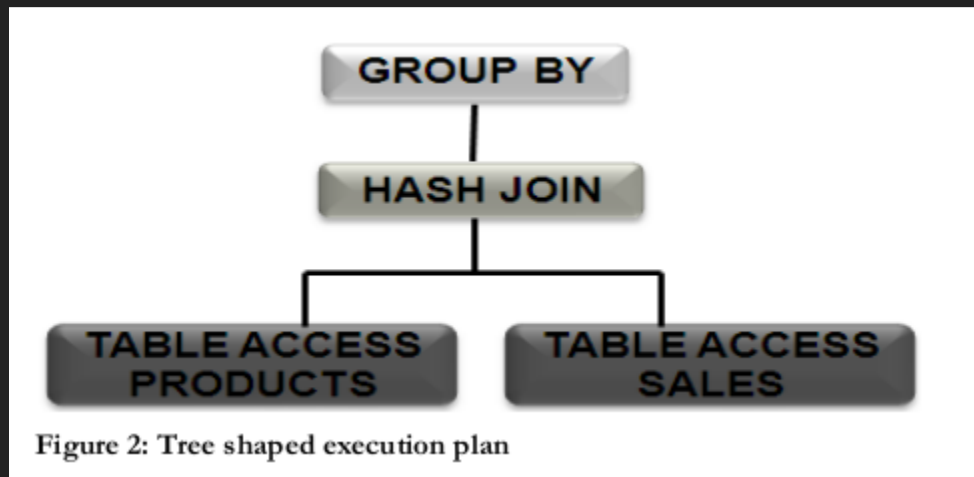
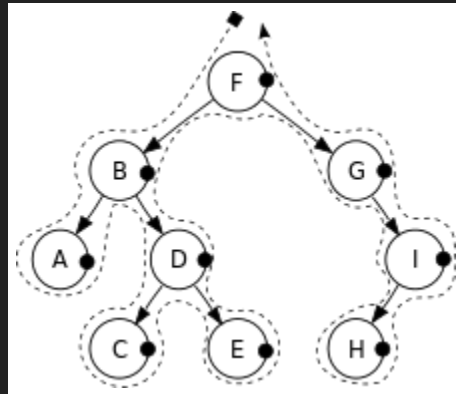


Figure 2: Tree shaped execution plan

트리구조에서 **Post-order traversal**의 순서를 따릅니다.

**A - C - E - D - B - H - I - G - F**



따라서 앞의 표에서 Execution plan의 순서를 ID순으로 나열하면

**3 - 5 - 4 - 2 - 1 - 0**

# COST란?

Query에 의해 사용된 IO, CPU, 네트워크 리소스를 점유한 정도를 가지고  
Oracle Optimizer가 정한 리소스 사용 기준으로,  
리소스가 가장 적다고 산정되는 Execution plan을 선택합니다.

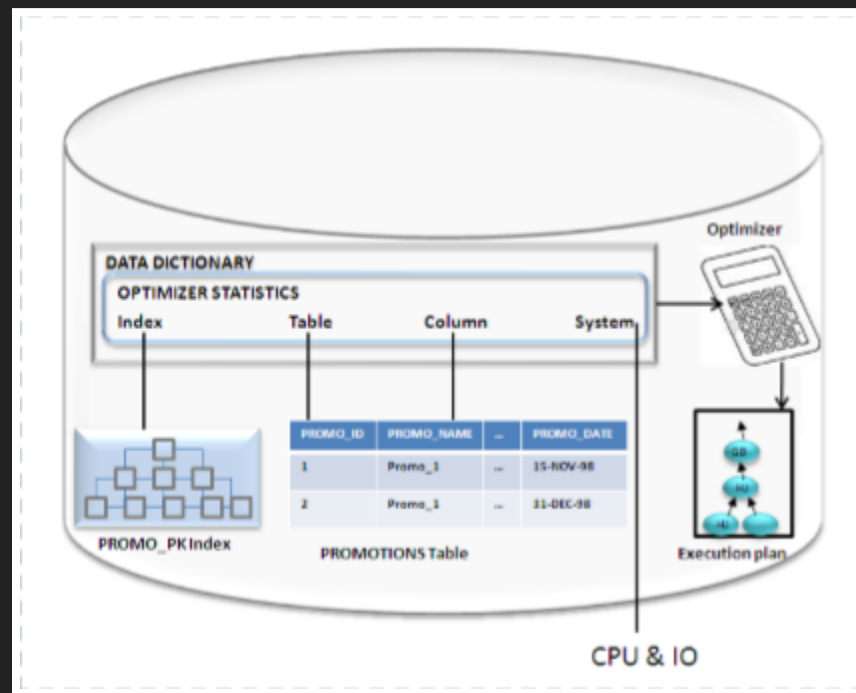
# OPTIMIZER가 EXECUTION PLAN을 결정하는 요소

Component	Description
Cardinality	각 Operation을 수행 시 반환되는 행의 수
Access method	데이터가 각 테이블 또는 인덱스에 접근하는 방법
Join method	테이블을 조인 시 사용되는 메소드(e.g., hash, sort-merge, etc.)
Join type	조인의 유형 (e.g., outer, anti, semi, etc.)
Join order	각 테이블이 조인되는 순서
Partition pruning	테이블이나 인덱스가 더 작은 조각으로 쪼개질 수 있는가 (Pstart Pstop 항목)
Parallel Execution	각 Operation이 병렬적으로 실행되었는가



# STATISTICS

Data Dictionary에 저장된 데이터의 모임  
USER\_TAB\_STATISTICS에 접근하여 볼 수 있다.



# STATISTICS

Statistic Name	Description
recursive calls	유저와 시스템 레벨에서 모두 생성된 재귀 호출 수.
db block gets	현재 블록이 요청된 횟수
consistent gets	요청된 블록을 일관적으로 읽은 횟수
physical reads	디스크로부터 읽은 총 데이터 블록 수
redo size	바이트로 생성된 리두의 총량
bytes sent via SQL*Net to client	포그라운드 프로세스에서 클라이언트로 전송된 총 바이트 수
bytes received via SQL*Net from client	Oracle Net을 통해 클라이언트로부터 받은 총 바이트 수
SQL*Net roundtrips to/from client	클라이언트로부터 주고 받은 Oracle Net의 모든 메시지들의 수
sorts (memory)	어떠한 디스크 쓰기도 처리하지 않아 완전히 메모리에서만 처리한 sort operation의 수
sorts (disk)	한번이라도 디스크 쓰기를 요구한 sort operation의 수
rows processed	작업 중에 처리 된 행의 수

# REFERENCE

- [\[PDF\] The Oracle Optimizer Explain the Explain Plan](#)
- [\[PDF\] Understanding Optimizer Statistics](#)
- [About SQL Processing](#)
- [The Query Optimizer](#)
- [Using EXPLAIN PLAN](#)
- [Using Autotrace in SQL\\*Plus](#)