

JPA SQL ,

SQL SQL

SQL JPA .

- JPA Query Method TO SQL
  - JPA-SQL Trace
- JPA SQL Trace
  - 
  -
- VS
  - 
  - 
  -
- /
  - VS
- JPA
  -

SQL , IO

DB

SQL , DataBase .

, JPA SQL

|           |                                   |
|-----------|-----------------------------------|
|           |                                   |
| Oracle    | set autotrace traceonly           |
| MSSQL     | SET SHOWPLAN_TEXT_ON              |
| DB2       | EXPLAIN ALL WITH SNAPSHOT FOR SQL |
| PostreSQL | EXPAIN SQL                        |
| MySQL     | EXPLAIN SQL                       |

JPA Query Method TO SQL

| JPA                                   | SQL                                                                                                 |
|---------------------------------------|-----------------------------------------------------------------------------------------------------|
| studentList = findByGroupInfoName("") | SELECT * FROM user ui left join spring.group_info gi<br>on gi.group_id = ui.group_id and gi.name="" |

JPA , ,

SQL , JPA .

, SQL .

JPA SQL ?

?( .) SQL

.

JPA-SQL Trace

- spring.jpa.properties.hibernate.show\_sql=true
- spring.jpa.properties.hibernate.use\_sql\_comments=true
- spring.jpa.properties.hibernate.format\_sql=true

JPA SQL Trace

JPA

TestCase SQL .

SQL ,

.

```
UserPageRepo.java  application.properties  Study1ApplicationTests.java
System.out.println( String.format(" %d:Contents %d:Page", sPage2.getNur

List<User> studentList = userPageRepo.findByGroupInfoName("학생");
System.out.println(studentList.get(0).getName());

}

<
Console  Progress  Problems
<terminated> Study1ApplicationTests [JUnit] C:\Program Files\Java\jdk1.8.0_144\bin\javaw.exe (2018. 1. 2
from
    user user0_
left outer join
    group_info groupinfo1_
        on user0_.group_id=groupinfo1_.group_id
where
    groupinfo1_.name=?
Hibernate:
select
    groupinfo0_.group_id as group_id1_0_0_,
    groupinfo0_.name as name2_0_0_,
    users1_.group_id as group_id4_1_1_,
    users1_.user_id as user_id1_1_1_,
    users1_.user_id as user_id1_1_2_,
    users1_.email as email2_1_2_,
    users1_.group_id as group_id4_1_2_,
    users1_.name as name3_1_2_
from
    group_info groupinfo0_
left outer join
    user users1_
        on groupinfo0_.group_id=users1_.group_id
where
    groupinfo0_.group_id=?
minsu22
```

Trace SQL , .

SQL JPA...,

.

user

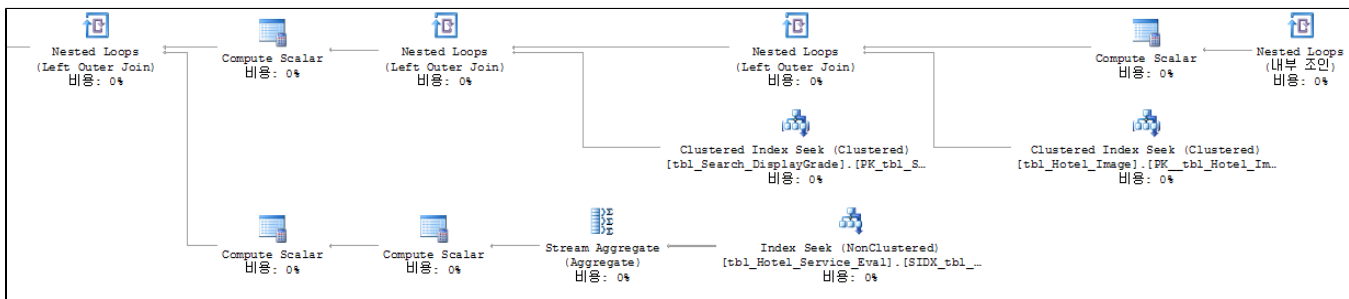
```

1 • USE spring;
2 • EXPLAIN EXTENDED
3 • SELECT * FROM user ui left outer join spring.group_info gi on gi.group_id = ui.group_id and gi.name="학생"
4

```

Result Grid

|  | id | select_type | table | partitions | type | possible_keys | key  | key_len | ref  | rows | filtered | Extra                                              |
|--|----|-------------|-------|------------|------|---------------|------|---------|------|------|----------|----------------------------------------------------|
|  | 1  | SIMPLE      | ui    | NONE       | ALL  | NONE          | NONE | NONE    | NONE | 102  | 100.00   | NONE                                               |
|  | 1  | SIMPLE      | gi    | NONE       | ALL  | PRIMARY       | NONE | NONE    | NONE | 2    | 100.00   | Using where; Using join buffer (Block Nested Loop) |



:DB Management Tool (MSSQL)

DB .

- 
- 
- 
- 

VS

```

@Entity
public class User {
    private String name;

    @ManyToOne
    @JoinColumn(name = "GROUP_ID", nullable = true)
    private GroupInfo groupInfo;
}

```

ManyToOne JPA

```

KEY `FKa36i4ekojwk70bxen390i6tek` (`group_id`) USING BTREE,
CONSTRAINT `FKa36i4ekojwk70bxen390i6tek` FOREIGN KEY (`group_id`) REFERENCES `group_info`
(`group_id`)

```

| JPA                 | SQL                                       |
|---------------------|-------------------------------------------|
| findByName("Minsu") | SELECT * FROM user where name = 'minsu2'; |

Name , minsu2

. ()

.

| JPA         | SQL                                    |
|-------------|----------------------------------------|
| findById(1) | SELECT * FROM user where group_id = 2; |

, .

, .

, .

.

SQL DBMS

.

Tip (Mysql Extra) .

Type

.

VS

|  | <div> <div> <div> <div>1페이지</div> <div>1</div> <div>2</div> <div>3</div> </div> <div> <div>2페이지</div> <div>4</div> <div>5</div> <div>6</div> </div> <div> <div>3페이지</div> <div>7</div> <div>8</div> <div>9</div> </div> <div> <div>4페이지</div> <div>10</div> <div>11</div> <div>12</div> </div> </div> <div> <div> <div>1</div> <div>4</div> <div>7</div> <div>10</div> </div> <div> <div>1페이지</div> <div>2페이지</div> <div>3페이지</div> <div>4페이지</div> </div> </div> </div> | <div> <div> <div>인덱스 1페이지</div> <div>1</div> <div>2</div> <div>3</div> </div> <div> <div>인덱스 2페이지</div> <div>4</div> <div>5</div> <div>6</div> </div> <div> <div>인덱스 3페이지</div> <div>7</div> <div>8</div> <div>9</div> </div> </div> <div> <div> <div>1</div> <div>4</div> <div>7</div> </div> <div> <div>인덱스 1페이지</div> <div>인덱스 2페이지</div> <div>인덱스 3페이지</div> </div> </div> <div> <div> <div>인덱스 1페이지</div> <div>1-1</div> <div>2-2</div> <div>1-3</div> </div> <div> <div>인덱스 2페이지</div> <div>3-1</div> <div>1-2</div> <div>2-3</div> </div> <div> <div>인덱스 3페이지</div> <div>2-1</div> <div>3-2</div> <div>3-3</div> </div> </div> |
|--|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|          |                                                                            |                                                                             |
|----------|----------------------------------------------------------------------------|-----------------------------------------------------------------------------|
|          | <ul style="list-style-type: none"><li>• 30%</li><li>• 7,8,9 IO 1</li></ul> | <ul style="list-style-type: none"><li>• 3%</li><li>• 7,8,9 2,3 IO</li></ul> |
|          | 1                                                                          | 249                                                                         |
| (DB<br>) | <del>PRIMARY KEY</del><br>: . PRIMARY KEY<br>...<br>( DB)                  |                                                                             |
|          | , Order                                                                    |                                                                             |
|          | <ul style="list-style-type: none"><li>•</li><li>•</li></ul>                | <ul style="list-style-type: none"><li>•</li><li>•</li></ul>                 |
|          |                                                                            |                                                                             |

/

Type 3 .

- (Nested Loops) :
- Sort Merge :
- Hash :

,

Sort Merge .( )

,

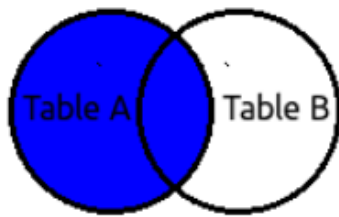
, .

- 
- 

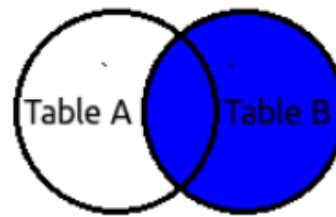
: <https://community.modeanalytics.com/sql/tutorial/sql-performance-tuning/>

,2 JOIN

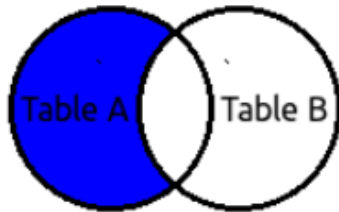
.



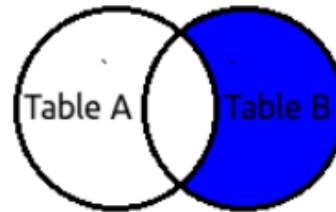
SELECT [list] FROM  
[Table A] A  
LEFT JOIN  
[Table B] B  
ON A.Value = B.Value



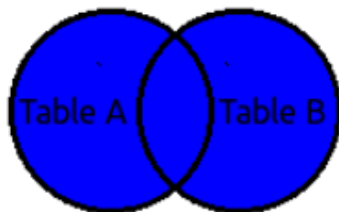
SELECT [list] FROM  
[Table A] A  
RIGHT JOIN  
[Table B] B  
ON A.Value = B.Value



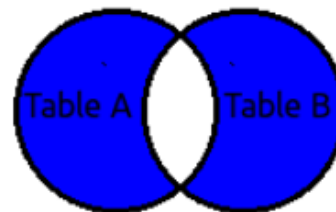
SELECT [list] FROM  
[Table A] A  
LEFT JOIN  
[Table B] B  
ON A.Value = B.Value  
WHERE B.Value IS NULL



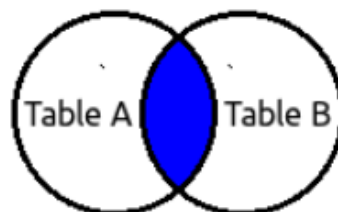
SELECT [list] FROM  
[Table A] A  
RIGHT JOIN  
[Table B] B  
ON A.Value = B.Value  
WHERE A.Value IS NULL



SELECT [list] FROM  
[Table A] A  
FULL OUTER JOIN  
[Table B] B  
ON A.Value = B.Value



SELECT [list] FROM  
[Table A] A  
FULL OUTER JOIN  
[Table B] B  
ON A.Value = B.Value  
WHERE A.Value IS NULL  
OR B.Value IS NULL



SELECT [list] FROM  
[Table A] A  
INNER JOIN  
[Table B] B  
ON A.Value = B.Value

## JPA

JPA ,

OOP, .

JPA ...

```
User user1 = findByName("Minsu");
User user2 = findByName("Minsu");
user1.GetName()
user2.GetName()
```

SQL/SP , Select DB .

JPA ,

SELECT SQL , DB .

DB IO , .

JPA , SQL .

SQL .

SQL DB IO Read

JPA

JPA .

SQL , .

Read

- ( Eager loading ) :
- ( Lazy loading ) :

- ALL :
- PERSIST :
- MERGE : merge()
- REMOVE :
- REFRESH :
- DETACH : DETACH, ENTITY DEATCH
- orphanRemoval :

JPA

.

SQL/SP

RDB JPA

.

, .



JPA . Select

SP .

( X )

Spring . DB ORM JPA .

Spring JPA ..., SQLMapper Mybatis .(Mybatis Spring )

- = Spring
- Spring DB = JPA,
- Spring ... ( ? )

.

Spring Mybatis

.

Mybatis . Mybatis -DB .

|          | Spring  | Asp.net            |
|----------|---------|--------------------|
| SQL/SP   | Mybatis | Enterprise Library |
| ORM      | JPA     | Entity framework   |
| Language | JAVA    | C#                 |

DB Asp.net Spring .

DB ? .

ASP.net MS DB ? .

...

Mybatis ORM , JPA .

MS SP .

ORM MS ORM , Spring JPA .

, ORM

Enterprise Library VS Mybatis .

SP VS .

ORM .

MS .

MS , SP

MS . ( )

, DB .

SP , .

MS JAVA ORM . -Entity FrameWork

ORM DB .

ORM .

- ORM . ==> TypeSafet , .
- ORM ==> DB . SQL2000 2018
- ORM . ==> QueryDSL SQL SQL , . . ORM . SP SQL SQL . ( . )
- ORM ERD - ==> ERD , . ERD , ORM ERD . ERD .

1:1 , JPA Inner Join ?  
JPA .  
InnerJoin  
JPA Null InnerJoin .  
DBMS ORM .  
JPA RDB  
ERD UML ,  
... , JPA .

DBMS SQL , SQL .

DBMS , DB .

ORM DB SQL , DB

VS ORM

? .

Table 3.8. Hibernate SQL Dialects (hibernate.dialect)

| RDBMS                     | Dialect                                     |
|---------------------------|---------------------------------------------|
| DB2                       | org.hibernate.dialect.DB2Dialect            |
| DB2 AS/400                | org.hibernate.dialect.DB2400Dialect         |
| DB2 OS390                 | org.hibernate.dialect.DB2390Dialect         |
| PostgreSQL                | org.hibernate.dialect.PostgreSQLDialect     |
| MySQL5                    | org.hibernate.dialect.MySQL5Dialect         |
| MySQL5 with InnoDB        | org.hibernate.dialect.MySQL5InnoDBDialect   |
| MySQL with MyISAM         | org.hibernate.dialect.MySQLMyISAMDialect    |
| Oracle (any version)      | org.hibernate.dialect.OracleDialect         |
| Oracle 9i                 | org.hibernate.dialect.Oracle9iDialect       |
| Oracle 10g                | org.hibernate.dialect.Oracle10gDialect      |
| Oracle 11g                | org.hibernate.dialect.Oracle10gDialect      |
| Sybase                    | org.hibernate.dialect.SybaseASE15Dialect    |
| Sybase Anywhere           | org.hibernate.dialect.SybaseAnywhereDialect |
| Microsoft SQL Server 2000 | org.hibernate.dialect.SQLServerDialect      |
| Microsoft SQL Server 2005 | org.hibernate.dialect.SQLServer2005Dialect  |
| Microsoft SQL Server 2008 | org.hibernate.dialect.SQLServer2008Dialect  |
| SAP DB                    | org.hibernate.dialect.SAPDBDialect          |
| Informix                  | org.hibernate.dialect.InformixDialect       |
| HypersonicSQL             | org.hibernate.dialect.HSQLDialect           |
| H2 Database               | org.hibernate.dialect.H2Dialect             |
| Ingres                    | org.hibernate.dialect.IngresDialect         |
| Progress                  | org.hibernate.dialect.ProgressDialect       |
| Mckoi SQL                 | org.hibernate.dialect.MckoiDialect          |
| Interbase                 | org.hibernate.dialect.InterbaseDialect      |
| Pointbase                 | org.hibernate.dialect.PointbaseDialect      |
| FrontBase                 | org.hibernate.dialect.FrontbaseDialect      |
| Firebird                  | org.hibernate.dialect.FirebirdDialect       |

DB .

- DDL(Data Definitison Language) (Create),(Alert),(Drop)
- DML(Data Manipulation Language) : Insert/Update/Select ( )
- DCL(Data Control Language) : Commit,RollBack,SavePoint (DB )

JPA-ORM DDL/DML/DCL .

ORM RDB . 3

... 3 .

, .

SP :

- SP MSSQL , DB / .
- SP , .
- DDL/DML/DCL .
- SP .

ORM :

- DML DCL ( Entity Proxy )
- SP , SP SP . .
- .
- DBMS SQL . ORM .

SP DB

.

.

, ORM RDB .