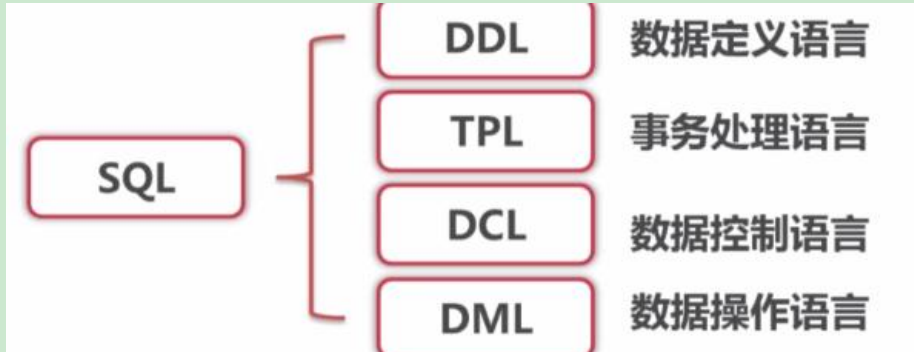


mysql开发技巧 1

JOIN的用法与分类（如何正确使用join语句） 如何实现分组选择数据

常见的sql语句类型



sql开发技巧侧重于DML语句
也就是select insert update delete

正确使用sql语句

增强数据库处理效率，减少应用相应时间
减少数据库服务器负载，增强服务器稳定性
减少服务器通讯的网络流量

框架虽然会自动生成sql语句，但是在因为通用，所以在效率上就没有我们自己写的效率高。

最重要实现的是我们无法通过框架无法实现的功能

如何正确的使用join从句

join 内连接 (Inner)

全外链接(Full outer)

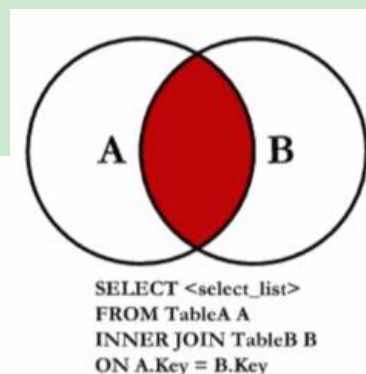
左外链接 (LEFT OUTER)

右外链接(RIGHT OUTER)

交叉链接(CROSS)

Join操作的类型—Inner Join

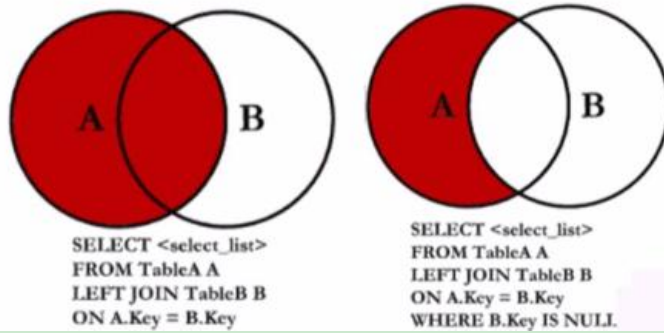
内连接Inner join基于连接谓词将两张表(如 A 和 B)的列组合在一起，产生新的结果表。



INNER JOIN的作用就是这样，将俩张表的公共部分显示出来

左外链接 (LEFT OUTER)

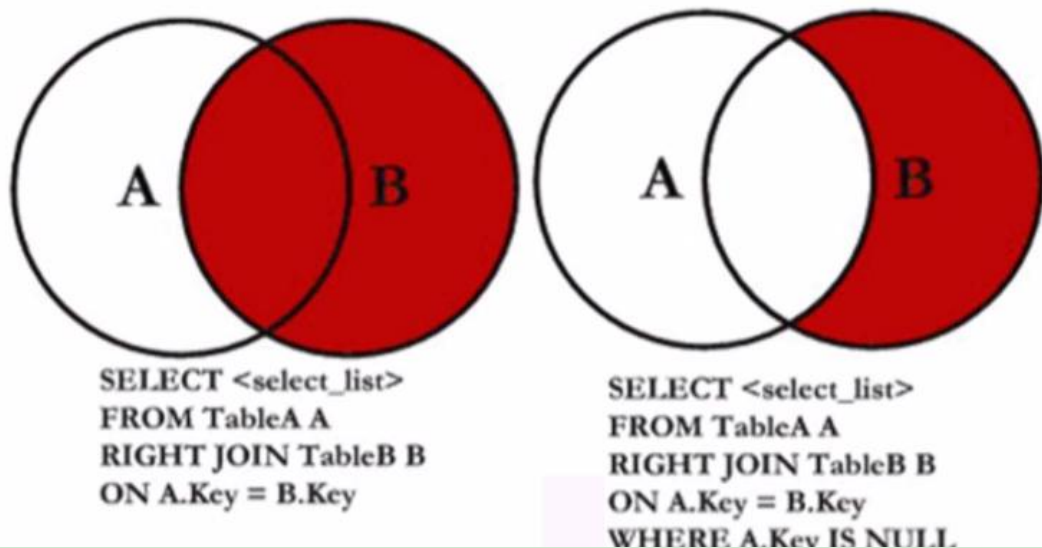
Join操作的类型—Left Outer Join



WHERE B.Key IS null B表中为null的记录 就是只查询A表中独特的记录
查询在A表中有, 在B表中没有的数据, 通常最简单的方法使用not in
not in 往往不会使用索引的

右外链接(RIGHT OUTER)

Join操作的类型—Right Outer Join

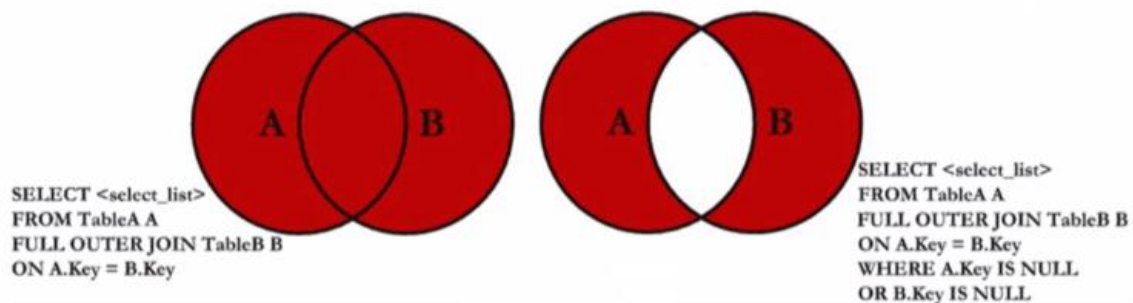


无论是left join 还是right join 都一定要考虑好是以哪个表为基础的

全外链接(Full outer)

实际就是左外连接与右外连接的合体

Join操作的类型—Full Join



直接在数据库中使用 full join会出现1064异常

Join操作的类型—Full Join 如何解决？

```
SELECT a.`user_name`,a.`over`,b.`over`  
FROM user1 a  
LEFT JOIN user2 b ON a.`user_name`=b.`user_name`  
UNION ALL  
SELECT b.`user_name`,b.`over`,a.`over`  
FROM user1 a  
RIGHT JOIN user2 b ON a.`user_name`=b.`user_name`
```

UNION ALL 是连接两个sql语句在一起的

交叉链接(CROSS JOIN)

在平常使用情况下，应避免这种笛卡尔集的查询，但是在某些特殊的情况下，或许是需要笛卡尔集的，来产生一点我们需要的笛卡尔集

Join操作的类型—Cross Join

交叉连接(cross join)，又称笛卡尔连接(cartesian join)或叉乘(Product),如果 A 和 B 是两个集合，它们的交叉连接就记为: $A \times B$.

Join操作的类型—Cross Join

```
SELECT a.`user_name`,a.`over`,b.user_name,b.`over`  
FROM user1 a  
CROSS JOIN user2 b
```

与join相关的sql技巧

如何更新使用过滤条件中包含自己的表

使用Join来解决问题

```
UPDATE user1 a JOIN (  
  SELECT b.`user_name`  
  FROM user1 a INNER JOIN user2 b ON  
a.`user_name`=b.`user_name`  
) b ON a.user_name=b.user_name  
SET a.over='齐天大圣'  
;
```

使用join优化子查询

优化前

使用JOIN优化子查询

```
SELECT a.user_name  
      ,a.`over`  
      ,(SELECT over FROM user2 b  
WHERE a.user_name=b.`user_name`) AS  
over2  
FROM user1 a ;
```

优化后

使用JOIN优化子查询

```
SELECT a.`user_name`,a.`over`,b.`over` AS over2
FROM user1 a
LEFT JOIN user2 b ON
a.`user_name`=b.`user_name` ;
```

使用join优化聚合子查询

使用JOIN优化聚合子查询

问题：如何查询出四人组中打怪最多的日期？

```
SELECT a.user_name,b.timestr,b.kills
FROM user1 a JOIN user_kills b
ON a.id = b.user_id
WHERE b.kills= ( SELECT MAX(c.kills) FROM user_kills c WHERE
c.user_id=b.user_id)
```

如何实现分组选择

如果我们有一些记录，这些记录可以分成多个分类，我们如果要在每个分类中，选出一定数量的数据。

如何实现分组选择

我们的场景

显然：

显然：

```
SELECT a.user_name,b.timestr,b.kills
FROM user1 a JOIN user_kills b
ON a.id = b.user_id
WHERE user_name = '孙悟空'
ORDER BY b.`kills` DESC
LIMIT 2
```

我们可以对每个人分别执行上面的查询

问题：1.如果分类或是用户很多的情况下则需要多次执行同一查询
2.增加应用程序同数据库的交互次数
3.增加了数据库执行查询的次数，不符合批处理的原则
4.增加了网络流量

```
WITH tmp AS (
SELECT a.user_name ,b.timestr ,b.kills
      ,ROW_NUMBER() over(partition by a.user_name order by
b.kills) cnt
FROM user1 a JOIN user_kills b
ON a.id = b.user_id
)select *
from tmp where cnt <=2
```

在SQL Server 或Oracle中我们可以这么做 甚至PgSQL 都可以

在mysql中不支持的 **不适用！**

如果在工作中用到分类聚合的方式，就不妨使用这种方式，一次性取出所有结果

```
SELECT d.user_name,c.timestr,kills
FROM (
SELECT user_id,timestr,kills
,(SELECT COUNT(*) FROM user_kills b WHERE
b.user_id=a.user_id AND a.kills<= b.kills) AS cnt
FROM user_kills a
GROUP BY user_id,timestr,kills
) c JOIN user1 d ON c.user_id=d.id
WHERE cnt <=2
```