

目录

1. 表连接有哪些？都是怎样连接的？	8
2. 什么是笛卡尔积连接？优缺点。	8
3. 如何查询出姓名重复？	8
4. A 表 123 三条数据,B 表 22 两条数据,左连接得到的数据是哪几条, 具体是怎么显示的	8
5. 列举常用的函数	9
6. 有一张表, 字段有学生姓名、课程名称和分数, 查出每门课程都 大于 90 分的学生姓名假设表叫 student	10
7. 索引	10
8. HIVE.....	11
9. 数据库的优化和提升技能.....	12
10. 主键、外键的看法	13
11. 遇到多层嵌套的 sql 语句, 有哪些优化方式?	13
12. 左连接, 右连接, 内连接, 全连接的区别是什么?	13
13. Substr 是什么函数	14
14. 具体的优化的手段是什么.....	14
15. 在表中其中一列建立索引, 根据这一列查询重复数据	15
16. union 和 union all	15
17. 主键和唯一索引的区别.....	16
18. 数据优化	16
19. 哪些地方用到了 SQL	17

20. dml 和 ddl 的区别	17
21. sql 优化问题如何处理	17
22. 优化和索引的关系	17
23. 索引的分类	17
24. 用 Oracle 干了什么;	17
25. 查询本月最后一天;	17
26. 建立索引需要注意什么	18
27. date 型和 时间戳型的区别是什么	18
28. where 和 having 的区别 是什么	18
29. 查看上一周的数据	19
30. 查看空值的方法	19
31. sql 语句注意什么?	19
32. 为什么用分析函数, 会提高效率吗?	19
33. 怎样进行逻辑删除和物理删除?	19
34. 索引要怎么用, 会提高效率么?	20
35. 索引什么时候用, 索引要注意什么? 缺点是什么-	20
36. 如何断定这条语句是高效的?	20
37. 数据库你都熟悉哪几个	20
38. 索引的优缺点?	21
39. oracle 的默认端口号是什么	21
40. 我在 oracle 中我怎么给一张表做备份	21
41. 将备份的表备份到其他数据库不用工具用 oracle 软件的命令 ..	21

42. 查询某张表的前几行/*不全部显示只显示一部分*/	21
43. 截取字符串、替换字符串、字符串转日期、日期转字符串	21
44. 监听的启动命令	22
45. 如何快速的清空一张表的数据	22
46. 数据库的启动命令	22
47. 数据库的停止命令	22
48. etl 加工时，a 表数据向 b 表插，两表可能会有主键冲突，从创建 时间字段，怎么取最新的那一条	22
49. oracle 里强制走全表扫描，不走索引怎么做，加哪些关键字	23
50. 学生 id 学科 id 成绩（记录条数不固定，取所有成绩大于 80 分的 学生 id 怎么做)除了用函数还有别的方式吗	23
51. 查看一个文件的倒数 50 行	23
52. submit 函数	23
53. 假如一个表有三个字段，怎么将他们在一个单元格中显示（字符 串合并）	23
54. sql 优化的时候怎么建索引	24
55. 数据都是重复的，用 delete 语句怎么删除一条	24
56. 索引概念	24
57. 索引优缺点，会让 oracle 变大吗	25
58. 列举 SQL 关联及平时运用方式，如何优化	25
59. 聚合函数	26
60. 字符串函数，日期截取，	26

61. 空值如何处理	26
62. 平时用的数据库是什么 oracle 的物理结构是什么	26
63. Oracle 的数据类型	27
64. 什么是事物	27
65. Oracle 里面的存储过程	27
66. 索引的应用	28
67. left join 和 inner join 的区别	29
68. oracle 和 mysql 之间的区别	29
69. 使用数据库时所用到的优化.....	29
70. 介绍一下索引	30
71. 索引分类，什么时候用到过位图索引，索引注意什么事项	30
72. select 查询语句慢怎么办	31
73. 索引：在什么情况下建索引.....	31
74. 以前用过 ETL 工具嘛？用过多久	31
75. 学号学科成绩，如何通过 ETL 去配置。	31
76. ETL 全拼是啥，代表啥意思？	31
77. 你熟悉 SQL 语句嘛？	32
78. drop,truncate,delete	32
79. sqlsever 与 oracle 区别	32
80. 关系型数据库	33
81. 我有一个表，里面有学号，学科，成绩三个字段，需要你算出成绩总分大于 600 分的学生以及他们的成绩，并且按照成绩从高到低排	

序，写出语句	33
82. Oracle 的导入导出命令	33
83. Hive 的分区表，内表，外表，数据倾斜处理，left join，map join， having 的用法，spark executor 的作用	34
84. SQL 优化储存	34
85. merge into.....	34
86. 查询家庭成员大于三且有两位年龄大于 80 岁的老人的户号	35
87. 代码打印九九乘法口诀.....	35
88. 函数练习题一.....	35
89. 用一条 sql 查询平均分，最高分，最低分，男生最高分，女生最 高分设有一个表 sc 是储存学生的所有信息，其中 ssex 字段储存学生 性别信息，score 字段储存学生的成绩信息；	36
90. 自定义函数	36
91. 写一个自定义函数，判断两个值的大小	37
92. 交集和并集	37
93. 怎么做行列转换	37
94. 关系型数据库了解多少，用过哪些.....	38
95. 讲解你认识的 ETL 和大数据	38
96. 索引什么情况下会失效.....	38
97. 哪些地方用到了 SQL	38
98. 去除重复数据	38
99. 怎么使用表连接去重.....	39

100. 1.地区编号 产品编号 销售量； 查询每个地区销售量前十的产品， 在平时的工作中写过类似的 sql 吗	39
101. A 表目前有 100 万数据， B 表 10 条数据， 根据关联键匹配， A 表中的数据与 B 表中的数据关联一一对应， A 表中能关联上的 48 万数据， B 表中能关联上的数据为 5 条。 那么问， 用 inner join, left join, right join, full join 的连接方式， 最终查询出多少数据量是多少。	39
102. 十道银行 oracle 试题， 口述	40
103. sql 语句 都使用过什么	40
104. 优化遵循的总体原则是什么	40
105. rowid 去除重复数据	40
106. 视图的作用	40
107. 有哪几种常见的索引	40
108. 索引需要注意的事项	41
109. 刚才说的在索引上使用 like 应该注意什么	41
110. 什么情况要用外连接	41
111. 开窗函数了解过吗？ 都知道些什么？	42
112. 主要使用什么数据库	42
113. 左连接与内连接有什么不同？	42
114. 给一张表， 有三列， 一列学生姓名， 一列课程名， 一个学生可以有多门课程， 一列成绩， 请问如何查询每个学生自己成绩大于 80 的所有科目？	42
115. 索引的使用；	42

116. 索引跟主键的区别	43
117. 创建一个索引，查询、分组、排序中哪个能用到索引	43
118. 表连接 ，左连接会显示多少条数据.....	43
119. sql 优化，exists 与 in.....	43
120. sql 优化，执行计划（主要看什么）	43
121. 表关联	43
122. 开窗函数，三种排数的区别.....	44
123. 怎么看这个字段是否用了索引.....	44
124. 在学校学过 MySQL，那工作的时候用过 Oracle 吗？会写存储过程吗？	44
125. 重复数据的去重以及重复数据的查找.....	44
126. Sql 4、5 种语言.....	44
127. 为什么会失效	45
128. 不会或者有歧义的题：15.97.99.127.....	45

1. 表连接有哪些？都是怎样连接的？

内连接

语法：FROM A [INNER] JOIN B ON A.a=B.b;

特点：按照关联条件关联，将两表中都符合关联条件的数据取出。

外连接

①左外连接：

语法：FROM A LEFT [OUTER] JOIN B ON A.a=B.b;

特点：按照关联条件关联，左表作为主表，将两表中都符合关联条件的数据以及主表中独有的数据取出。

②右外连接：

语法：FROM A RIGTH [OUTER] JOIN B ON A.a=B.b;

特点：按照关联条件关联，右表作为主表，将两表中都符合关联条件的数据以及主表中独有的数据取出。

③全外连接：

语法：FROM A FULL[OUTER] JOIN B ON A.a=B.b;

特点：按照关联条件关联，将两表中都符合关联条件的数据以及左表中独有的数据和右表中独有的数据取出。

笛卡尔连接

语法：FROM A CROSS JOIN B;

特点：左表中的每一项数据都与右表中的每一条数据连接一次，数据量较大 不建议使用。

2. 什么是笛卡尔积连接？优缺点。

笛卡尔积连接就是将左表中的每一条数据都与右表中的每一条数据做一次连接，得到所有数据。

优点：不会造成数据的丢失，数据比较全面。

缺点：比较危险，容易造成数据的暴涨，给数据库带来宕机。

3. 如何查询出姓名重复？

以姓名分组，对分组后的姓名列进行计数，大于一的即重复姓名。

4. A 表 123 三条数据，B 表 22 两条数据，左连接得到的数据是哪几条，具体是怎么显示的

A 表左连接 B 表，得到的是以 A 为主表，显示 A 表和 B 表共有的数据和 A 所独有的数据

B 表左连接 A 表，得到的是以 B 为主表，显示 A 表和 B 表共有的数据和 B 所独有的数

据

5. 列举常用的函数

1. 聚合函数

- 1.1 sum() 求和
- 1.2 count() 计数
- 1.3 avg() 求平均值
- 1.4 max() 求最大值
- 1.5 min() 求最小值

2. 转换函数

- 2.1 转字符 to_char()
- 2.2 转数值 to_number()
- 2.3 转日期 to_date()

3. 字符型函数

- 3.1 连接函数 concat(str1,str2)
- 3.2 大小写转换 转大写 upper()
转小写 lower()
首字母大写 initcap()
- 3.3 替换函数 replace(str,s1,s2)
- 3.4 去除函数 trim() 去除字符串左右两边的空格
ltrim() 去除字符串左边的空格
rtrim() 去除字符串右边的空格
trim([leading/trailing/both] [s from] str)
- 3.5 截取函数 substr(str,ind,[,len])
从 str 的 ind 位置，截取 len 长度的字符串并返回该截取内容
- 3.6 填充函数 lpad(str,n,s)
rpad(str,n,s)
- 3.7 获取字符长度 length(str)
- 3.8 获取字符位置 instr(str,s [,ind,n])
从第 ind 位开始，查找 s 在 str 中第 n 次出现的位置

4. 数值型函数

- 4.1 取绝对值 abs()
- 4.2 向上取值 ceil()
向下取值 floor()
- 4.3 取余 mod(x,y)
- 4.4 四舍五入 round(number [,p])
- 4.5 截断 trunc(number [,p])

5. 日期型函数

5.1 获取月份的差值 months_between(d1,d2)

5.2 获得月末日期 last_day()

5.3 月份加减 add_months(d,n) 在 d 日期的基础上加 n 月

5.4 获取下个周几 next_day(d,w)

6. 其他函数

6.1 条件取值 decode(expr,value1,result1,...,def_result)

6.2 空值的赋值 nvl(expr,value)

6.3 去重 distinct col_list

6. 有一张表，字段有学生姓名、课程名称和分数，查出每门课程都大于 90 分的学生姓名假设表叫 student

```
Select  ename  from student group by ename having min(score)>90;
```

7. 索引

一、按存储形式

1.B-tree （Oracle 默认索引类型）

Create index ind_name on 表名(字段名)

2 位图索引

create bitmap index ind_name on 表名（字段名）

3 反向键索引

create index ind_name on 表名（字段名） reverse

4 基于函数的索引（原值已经建了索引，但加上函数后就失效了这时候需要用基于函数的索引）

Create index ind_name on 表名（函数（字段名））

二、按唯一性

1.唯一索引

Create unique index ind_name on 表名(字段名)

2.非唯一索引

Create index ind_name on 表名(字段名)

三、按列的个数

1.单列索引

create index ind_name on 表名（字段名）

2.复合索引

create index ind_name on 表名（字段名 1， 字段名 2）

四、索引的删除

Drop index ind_name

五、索引的禁用与重建

禁用: `alter index ind_name unusable`
重新启用: `alter index ind_name rebuild`
总结:
查询快, 插的慢。

8. HIVE

定义:

hive 是基于 Hadoop 的一个数据仓库工具, 用来进行数据提取、转化、加载, 这是一种可以存储、查询和分析存储在 Hadoop 中的大规模数据的机制。

hive 数据仓库工具能将结构化的数据文件映射为一张数据库表, 并提供 SQL 查询功能, 能将 SQL 语句转变成 MapReduce 任务来执行。

Hive 的优点是:

1. 学习成本低, 可以通过类似 SQL 语句实现快速 MapReduce 统计, 使 MapReduce 变得更加简单, 而不必开发专门的 MapReduce 应用程序。
2. hive 十分适合对数据仓库进行统计分析。
3. hive 的最佳使用场合是大数据集的批处理作业, 例如, 网络日志分析。 [4]

局限性:

1. hive 并不能够在大规模数据集上实现低延迟快速的查询, 例如, hive 在几百 MB 的数据集
2. hive 并不适合那些需要高实时性的应用, 例如, 联机事务处理 (OLTP)。
3. 并不提供实时的查询和基于行级的数据更新操作。

结构:

1. 用户接口 常用: cli; 还有 CLIENT 和 WUI

2. 元数据存储 mysql、derby。hive 中的元数据包括表的名字, 表的列和分区及其属性, 表的属性 (是否为外部表等), 表的数据所在目录等。

3. 解释器、编译器、优化器、执行器

解释器、编译器、优化器完成 HQL 查询语句从词法分析、语法分析、编译、优化以及查询计划的生成。

生成的查询计划存储在 HDFS 中, 并在随后由 MapReduce 调用执行。

4. Hadoop

hive 的数据存储在 HDFS 中,

大部分的查询由 MapReduce 完成 (不包含 * 的查询, 比如 `select * from tbl` 不会生成 MapReduce 任务)

存储模型:

hive 中包含以下四类数据模型：表(Table)、外部表(External Table)、分区(Partition)、桶(Bucket)。

9. 数据库的优化和提升技能

SQL 性能优化

- 1.尽量是有列名代替*
- 2.尽量使用 where 代替 having
- 3.尽量使用多表查询代替子查询
- 4.尽量使用绑定：SELECT * FROM EMP WHERE DEPTNO=&NUM;

性能优化法则及方式：

1.减少磁盘访问

减少数据访问

- 创建并使用正确的索引
- 只通过索引访问数据
- 优化 SQL 执行计划

2.减少网络传输

- 1) 返回跟更少数据（减少网络传输或磁盘访问）

- 数据分页处理
- 只返回需要的字段

- 2) 减少交互次数

BATCH DML

IN LIST

设置 FETCH SIZE

使用存储过程

优化业务逻辑

使用 RESULTSET 游标处理记录

3.减少数据库 CPU 运算

- 使用绑定变量
- 合理使用顺序
- 减少比较操作
- 大量复杂运算在客户端处理

4. 利用更多资源

- 客户端多进程并行访问
- 数据库并行处理

数据库优化的内容：

- 1.调整数据结构的设计
- 2.调整应用程序的设计
- 3.调整数据库 SQL 语句
- 4.调整服务器内存分配

- 5.调整硬盘 I/O，这步是在系统开发之前完成的
- 6.调整操作系统参数

10.主键、外键的看法

主键：

字段中既不能出现空值，又不能出现重复值

- 1.一个表中只能存在一个主键约束
- 2.建立了主键约束的字段，会自动建立一个唯一索引

外键：

外键是指引用另外一个表中的一列或多列数据，被引用的列应该具有主键约束或者唯一性约束。外键用来建立和加强两个表数据之间的连接。

- 1.以当前表作为子表，用来获取取值范围的另一张表作为主表来说，外键约束是建立在子表上的，而非主表
- 2.外键针对的是非空值，即空值不受外键约束限制
- 3.想要在子表上建立外键约束指向主表的字段，主表上被指定的字段必须是建立过主键约束或唯一约束
- 4.子表中若存在主表中不存在的数据，则试图建立外键时会失败
- 5.子表中想要插入或修改数据，处理后的数据必须存在于主表中，否则会处理失败
- 6.主表中已存在于子表中的内容，想要删除或修改时必须先删除或修改对应的子表内容，否则会处理失败

外键因其独特而顽固的限制，很少在项目中被使用，甚至有些项目组会禁止使用外键

11.遇到多层嵌套的 sql 语句，有哪些优化方式？

- 1.可以用连接查询方式来代替嵌套查询
- 2.建立有效的索引是程序运行中提高查询效率的有效方法。

12.左连接，右连接，内连接，全连接的区别是什么？

内连接：按照关联条件关联，将两表中都符合关联条件的数据取出

左外连接：按照关联条件关联，左表作为主表，将两表中都符合关联条件的数据以及主表中独有的数据取出

右外连接：按照关联条件关联，右表作为主表，将两表中都符合关联条件的数据以及主表中独有的数据取出

全外连接：按照关联条件关联，将两表中都符合关联关系的数据以及左表中独有的、右表中独有的数据取出

13.Substr 是什么函数

截取函数

SUBSTR(STR,IND[,LEN]):从字符串 STR 的 IND 位置,截取 LEN 长度的字符串并返回该截取内容

•总结:

- 1) 截取函数中的参数 IND 和 LEN 分别表示起始位置和截取长度,而非起始位置和终止位置;
- 2) 从 0 位开始截取,相当于从 1 位开始截取;
- 3) 如果 IND 参数为正,表示从左向右定位;如果 IND 参数为负,表示从右向左定位;
- 4) 无论 IND 参数是正还是负,截取时都是从左向右截取;
- 5) 若 LEN 参数为 0 或负数,截取结果返回空值;
- 6) 若 LEN 参数已超出 STR 实际长度,则截取到末尾;
- 7) LEN 参数可以省略,表示截取到末尾,而 IND 参数必不可少。

14.具体的优化的手段是什么

数据库常见的优化手段分为三个层面:SQL 和索引优化、数据库结构优化、系统硬件优化等

1) SQL 和索引优化

正确使用索引,索引是数据库中最重要概念之一,也是提高数据库性能最有效的手段之一,

它的诞生本身就是为了提高数据查询效率的,就像字典的目录一样,通过目录可以很快找到相

关的内容。我们应该尽可能的使用主键查询,而非其他索引查询,因为主键查询不会触发回表

查询,因此节省了一部分时间,变相的提高了查询的性能。

假如我们没有添加索引,那么在查询时就会触发全表扫描,因此查询的数据就会很多,并且查

询效率会很低,为了提高查询的性能,我们就需要给最常使用的查询字段上,添加相应的索引,

这样才能提高查询的性能

2) 数据库结构优化

① 最小数据长度,② 使用最简单数据类型,③ 尽量少定义 text 类型,④ 选择合适的存储引擎,⑤ 适当分表、分库策略

3) 系统硬件优化

① 磁盘

磁盘寻道能力(磁盘 I/O),以目前高转速 SCSI 硬盘(7200 转/秒)为例,这种硬

盘理论上每秒寻道 7200 次，这是物理特性决定的，没有办法改变;磁盘应该尽量使用有高性能读写能力的磁盘，比如固态硬盘，这样就可以减少 I/O 运行的时间，从而提高了 MySQL 整体的运行效率。

磁盘也可以尽量使用多个小磁盘而不是一个大磁盘，因为磁盘的转速是固定的，有多个小磁盘就相当于拥有多个并行运行的磁盘一样。

② 网络

保证网络宽带的通畅（低延迟）以及够大的网络带宽是 MySQL 正常运行的基本条件，如果条件允许的话也可以设置多个网卡，以提高网络高峰期 MySQL 服务器的运行效率。

DNS 配置 尽量使用 skip-name-resolve 来减少因解析带来的不必要麻烦。

检查网络的 ping 丢包率。

通过优化/etc/sysctl.cnf 中的网络参数，提升性能。

③ 内存

MySQL 服务器的内存越大，那么存储和缓存的信息也就越多，而内存的性能是非常高的，从而提高了整个 MySQL 的运行效率。

15.在表中其中一列建立索引，根据这一列查询重复数据

```
create index ind1 on emp(ename);
```

```
select * from emp where ename in (select ename from emp group by ename having count(1)>1);
```

16.union 和 union all

UNION:将两个查询结果进行合并，若存在相同的数据，将数据去重排序，结果以第一个字段升序显示

UNION ALL:将两个查询结果合并，且不会对重复数据进行去重及排序

注意：

- 1.两个合并的查询结果列名可以不一致，但列数、数据类型要保持一致
- 2.最终的列名和第一个查询结果中的列名相同
- 3.集合运算没有先后之分，依次从上到下执行，除非有括号
- 4.ORDER BY 排序只能放在最后一个查询内，列名不一致时候，可以使用序号或者别名

17.主键和唯一索引的区别

1.

主键一定是唯一性索引，唯一性索引并不一定就是主键

所谓主键就是能够唯一标识表中某一行的属性或属性组，一个表只能有一个主键，但可以有多个候选索引。

因为主键可以唯一标识某一行记录，所以可以确保执行数据更新、删除的时候不会出现张冠李戴的错误。

主键除了上述作用外，常常与外键构成参照完整性约束，防止出现数据不一致。

数据库在设计时，主键起到了很重要的作用。

主键可以保证记录的唯一和主键域非空,数据库管理系统对于主键自动生成唯一索引，所以主键也是一个特殊的索引。

2.

一个表中可以有多个唯一性索引，但只能有一个主键

3.

主键列不允许空值，而唯一性索引列允许空值

18.数据优化

数据库是支持知识发现的基础工程。

要获得高信息含量的、有用的知识，理想情况是原始数据是不含噪声的正确数据。

数据是组成数据库的基本单元，构建高质量的数据库，必须对原始数据进行数据预处理，也就是所说的数据优化（EVDO）。

优化流程

主要流程分三个方面，分别是数据采集、数据分析和数据处理。下面将分别介绍这三个步骤所要完成的主要功能。

1. 数据采集

软件测试方案生成子系统根据不同的测试部门和不同的测试项目的具体要求确定相应的数据采集范围，以实时数据和历史数据作为基础采集系统运行当中需要的数据。

2. 数据分析

对于实时数据，系统将根据输入的文件或者命令，将其按照给定的设计要求保存到对应的数据变量和数组中，供测试管理人员生成新方案使用。

对于历史数据，系统需根据不同的数据表内容，分析其中的有用信息，并对其进行深入的分析。

3. 数据处理

每次系统运行开始时，对系统中的各种数据进行总结与整理，将其按照系统要去放入对应的变量或者数据表当中，以供下一步工作使用

19. 哪些地方用到了 SQL

20. dml 和 ddl 的区别

dml(data manipulation language)是数据操纵语言，主要用于对数据库中的数据（行）进行增删改查操作，执行 dml 语句进行的操作不会直接进入数据库，需要手动提交，也可以回滚；ddl(data definition language)是数据定义语言，主要用于定义和修改表结构，数据类型，链接和约束，执行 ddl 语句进行的操作会被隐式提交，不可回滚；

21. sql 优化问题如何处理

- 1.创建合适的索引，（例如使用较多的筛选和排序条件）
- 2.分表：当一张表的数据比较多或一张表的某些字段的值比较多时，采用水平分表和垂直分表进行优化
- 3.读写分离：通过禁用索引和约束来提高操作数据的效率，变多次事务提交为一次提交，使用批量操作

22. 优化和索引的关系

建立索引是优化数据库的方式之一，可以在一定程度上提高数据库的查询效率。但使用索引并非必选的优化方式，要根据数据情况和优化需求合理使用索引。

23. 索引的分类

b-tree 索引，位图索引，反向键索引，函数索引

24. 用 Oracle 干了什么；

25. 查询本月最后一天；

```
select trunc(last_day(sysdate), 'dd') from dual;
```

26. 建立索引需要注意什么

1. 如果对某大表进行筛选，某列或者某几列频繁出现的 WHERE 子句，应该考虑在这些列上加索引
2. 如果对某大表进行排序时，某列或者某几列频繁出现在 ORDER BY 子句中，应该考虑在这些列上加索引
3. 小表不要建立索引
4. 对于含有空值的列，如果经常在查询时查询非空值，应该在该列上建立索引，如果经常在查询时查询空值，应该在该列上建立基于函数的索引 NVL()
5. 为了提高表连接的性能，应该在连接列上建立索引（建立普通索引即可）
6. 索引时数据库的一种实体对象，级别类似表，会占内存空间，ORACLE 会自动进行索引维护，表和索引建立在不同的表空间 手动维护：ALTER INDEX 用户名.索引名 COALESCE;
7. 通过索引可以提升数据的查询速度，但是会相对地降低 DML 语句的操作速度，尤其是插入的速度，ORACLE 会花费时间在索引维护上，所以要把握索引的数量
8. 对于复合索引，至少引用索引列的第一个参数才会使用该索引
9. 某列可以出现在多个索引中，但相同的某列或者某几列无法多次建立索引(A,B)与 (B,A)不同
11. 索引建立后不一定使用，ORACLE 会分析整个 SQL 语句的最优执行方式
12. ORACLE 会自动在主键约束和唯一约束建立唯一索引
13. 对于一般的 B-TREE 索引，通配符出现在首位时不会使用索引 LIKE '%A'（检索全部数据）
14. 在索引列上使用 <> != 不等号时，或对空值进行判断索引列不会生效(检索全部数据)

27.date 型和 时间戳型的区别是什么

date 类型的最小时间间隔是秒，当连续写入数据时不能通过 date 类型时间判断写入的先后顺序；

timestamp 可以存储世纪、4 位年、月、日、时、分、秒。与 DATE 类型相比，timestamp 可以存储毫秒和时区

28.where 和 having 的区别 是什么

where 和 having 都根据条件对表中的数据进行筛选

where: 对原始表的筛选， 不能使用聚合函数作为条件

having: 对分组后的表过滤筛选，可以使用聚合函数作为条件

29. 查看上一周的数据

使用 `trunc(sysdate, 'iw')` 得出当前周第一天的日期，将该日期减一得到上周最后一天的日期 `d`，再使用 `trunc(d, 'iw')` 的到上周第一天的日期 `D`，以 `[D-d]` 这一日期区间作为条件查询数据表，就能得到上周的数据

30. 查看空值的方法

条件查询，查询条件为 `where col_name is null;`

31. sql 语句注意什么？

sql 语句 本身没有 ; 分号是方便执行多行的时候一起执行区别语句
取别名时候,给字段取别名: 字段名 (as) 别名 ;给表取别名中间一定不能用 as
null 值和数字运算还是 null,null 不参与运算
SQL 不注重大小写,但是字段内容是注重大小写的
null 不能使用条件判断, 只能使用 is
单行函数: 一条记录返回一个结果
多行函数 组函数 聚合函数: 多条记录 返回一个结果
子查询可能返回一条记录, 也可能返回多条

32. 为什么用分析函数，会提高效率吗？

由于使用普通的聚合函数查询会受到分组条件的限制，只能基于分组条件显示数据，因此需要将分析函数与聚合函数相结合成窗口函数来满足特定的查询需求。

分组函数能提高编码效率和代码执行效率，如不使用分析函数，必须使用自联查询，子查询或者内联视图，甚至复杂的存储过程实现的语句。

33. 怎样进行逻辑删除和物理删除？

逻辑删除的本质就是修改操作，所谓的逻辑删除并不是真的删除，而是在表中将对应的是否删除标识(`is_delete`)或者说是状态字段(`status`)做修改操作。比如 0 是未删除，1 是删除。在逻辑上数据是被删除了，但是数据本身依旧存在库里。

物理删除就是真正的从数据库中删除操作，使用 `dml` 语句进行删除数据

34.索引要怎么用，会提高效率么？

拥有建立索引的权限的用户可以在数据表的字段上建立索引，语法为：`create index on table_name(column_name);`

当查询时，若筛选条件或排序依据命中索引，则可一定程度上提高查询效率。

35.索引什么时候用，索引要注意什么？缺点是什么-

某个列经常被作为筛选条件和分组排序的依据时应在该列建立索引
使用索引时应注意：

- 1.根据筛选条件的过滤强度选择是否使用索引，若返回大量值则索引的作用将收到限制
- 2.根据所选字段的数据情况建立合适的索引类型
- 3.当对多字段创建联合索引时，注意查询条件的顺序是否能命中索引

索引的缺点：

- 1.占用物理空间
- 2.降低 dml 的执行效率

36.如何断定这条语句是高效的？

- 1.查询优化器为当前的 SQL 语句生成最佳的执行计划，保证数据读写使用最佳路径；
- 2.设置合理的物理存储结构，如表的类型，字段的顺序，字段的数据类型等。

在 oracle 中我怎么给一张表做备份（离线备份）

- 1.使用数据字典找出所有需要备份的文件（控制文件，重做日志文件，数据文件及与表空间的对应关系，临时文件及对应关系，正文参数文件或二进制参数文件）
- 2.正常关闭数据库（使用 `shutdown immediate|transactional|normal`）
- 3.将所有文件复制到备份硬盘或磁带上
- 4.重新启动数据库（`startup`）

37.数据库你都熟悉哪几个

mysql: 关系型数据库管理系统，体积小，成本低，开源，最好的 web 应用 rdbms，可以和 PHP 搭配组成良好的开发环境。

oracle: 关系型数据库管理系统，移植性好，功能强大，使用大中小微各种计算机环境，高效，高可靠性，适应高吞吐量数据库解决方案。

sql server: 关系型数据库管理系统，为分布式客户机/服务器所设计的数据库管理系统，可扩展，高性能，与 windowsNT 有机结合。

db2: 关系型数据库管理系统，应用于所有常见的服务器操作平台，主要包括 unix, linux, z/os 以及 Windows 的服务器版本。

38. 索引的优缺点？

优点：加快数据检索的速度，提高检索效率

缺点：1.创建索引会占用物理存储空间 2.使用索引会牺牲 dml 语句的效率，因为对数据的增删改操作时需要动态维护索引；

39. oracle 的默认端口号是什么

1521

40. 我在 oracle 中我怎么给一张表做备份

创建表：myTable 的备份表 myTable_bak
create table myTable_tmpe as select * from myTable ;

41. 将备份的表备份到其他数据库不用工具用 oracle 软件的命令

先将备份表中的数据导出：\$ exp user/pwd file=/dir/xxx.dmp log=xxx.log tables=table_bak;
再将导出的数据导入新的数据库：\$ imp user/pwd file=/dir/xxx.dmp log=xxx.log tables=table_bak from user=dbuser touser=dbuser2 commit=y ignore=y

42. 查询某张表的前几行/*不全部显示只显示一部分*/

使用条件查询表中所有字段，条件为： rownum < n(n > 1)

43. 截取字符串、替换字符串、字符串转日期、日期转字符串

使用函数 substr(str, S, L), str 是被截取的字符串， s 是开始截取的位置， l 是截取的字符串长度

使用函数 replace(str, s1, s2), str 是被替换的字符串，s1 是字符串中要被替换的部分字符串，

s2 是用来进行替换的新字符串

使用函数 `to_date(str, format)`, `str` 是被转换的字符串, `format` 是转换后的日期的格式

使用函数 `to_char(date, format)`, `date` 是被转换的日期, `format` 是转换后的字符串的格式

44. 监听的启动命令

```
lsnrctl start
```

45. 如何快速的清空一张表的数据

```
truncate table tab_name;
```

```
delete from table_name;
```

46. 数据库的启动命令

1. 首先用 `SYSDBA` 的连接身份, 启动 `SQL*Plus` 并同时登录、连接到数据库。

```
sqlplus / as sysdba
```

2. `STARTUP`

47. 数据库的停止命令

1. 首先用 `SYSDBA` 的连接身份, 启动 `SQL*Plus` 并同时登录、连接到数据库。

```
sqlplus / as sysdba
```

2. `SHUTDOWN` 命令语法

```
SHUTDOWN [NORMAL | TRANSACTIONAL | IMMEDIATE | ABORT]
```

48. etl 加工时, a 表数据向 b 表插, 两表可能会有主键冲突, 从创建

时间字段, 怎么取最新的那一条

创建一个中间表, 无主键 `unique`。把 a, b 表日期数据都插进去。

```
select 中间表 groupby 主键, order by 时间字段 desc
```

然后将最新的数据 `insert` 到表格中

49. oracle 里强制走全表扫描，不走索引怎么做，加哪些关键字

```
SELECT /*+FULL(TB_NAME) */* FROM
```

```
SELECT /*+INDEX(TB_NAME IN_NAME) */* FROM
```

50. 学生 id 学科 id 成绩（记录条数不固定，取所有成绩大于 80 分的学生 id 怎么做）除了用函数还有别的方式吗

```
1. select name from table name
WHERE name not in (
select name from stu
where score <80)
```

51. 查看一个文件的倒数 50 行

```
1. 伪列实体化
select * from
(select e.*,rownum r from table_name )
where r between (select count(*)-49 from table_name) and (select count(*) from table_name);
```

52. submit 函数

53. 假如一个表有三个字段，怎么将他们在一个单元格中显示（字符串合并）

1. 使用合并字符串 "||";
2. 使用连接函数 CONCAT;

54.sql 优化的时候怎么建索引

索引建立或使用的规则与建议

1. 如果对某大表进行筛选，某列或者某几列频繁出现的 WHERE 子句，应该考虑在这些列上加索引
2. 如果对某大表进行排序时，某列或者某几列频繁出现在 ORDER BY 子句中，应该考虑在这些列上加索引
3. 小表不要建立索引
4. 对于含有空值的列，如果经常在查询时查询非空值，应该在该列上建立索引，如果经常在查询时查询空值，应该在该列上建立基于函数的索引 NVL()
5. 为了提高表连接的性能，应该在连接列上建立索引（建立普通索引即可）
6. 索引时数据库的一种实体对象，级别类似表，会占内存空间，ORACLE 会自动进行索引维护，表和索引建立在不同的表空间 手动维护：ALTER INDEX 用户名.索引名 COALESCE;
7. 通过索引可以提升数据的查询速度，但是会相对地降低 DML 语句的操作速度，尤其是插入的速度，ORACLE 会花费时间在索引维护上，所以要把握索引的数量
8. 对于复合索引，至少引用索引列的第一个参数才会使用该索引
9. 某列可以出现在多个索引中，但相同的某列或者某几列无法多次建立索引(A,B)与 (B,A)不同
11. 索引建立后不一定使用，ORACLE 会分析整个 SQL 语句的最优执行方式
12. ORACLE 会自动在主键约束和唯一约束建立唯一索引
13. 对于一般的 B-TREE 索引，通配符出现在首位时不会使用索引 LIKE '%A'（检索全部数据）
14. 在索引列上使用 <> != 不等号时，或对空值进行判断索引列不会生效(检索全部数据)

55.数据都是重复的，用 delete 语句怎么删除一条

```
1.delete FROM TB_NAME
  where rowid NOT in
      (select max(rowid) from TB_NAME group by COL_NAME1,...);
2.DELETE
  from (select T.*,
              row_number() over(partition by col_1.... order by 1) rn
        from EMP1 T) t1
 where t1.rn > 1;
```

56.索引概念

索引是一种供服务器在表中快速查找数据的数据库结构，是建立在表的一列或多列上的辅助对象。可以加快对表的访问速率。一种类似书的目录结构，占用磁盘空间。

57.索引优缺点，会让 oracle 变大吗

索引优点：

- 1) 大大加快数据的检索速度
- 2) 创建唯一性索引，保证数据库表中每一行数据的唯一性
- 3) 加速表与表之间的连接，特别是在实现数据的参考完整性方面特别有意义
- 4) 在使用分组和排序子句进行数据检索时，同样可以显著减少查询中分组和排序的时间

索引缺点：

- 1) 索引需要占物理空间
- 2) 当对表中的数据进行增加、删除和修改的时候，索引也要动态的维护，降低了数据的维护速度

会让 oracle 变大吗

会，索引是数据库的一种实体对象，级别类似于表，会占用内存空间，ORACLE 会自动进行索引维护，

表和索引可以建立在不同的表空间。

58.列举 SQL 关联及平时运用方式，如何优化

内连接：FROM A [INNER] JOIN B ON A.a = B.b;

外连接：左外连接：FROM A LEFT [OUTER] JOIN B ON A.a = B.b;

右外连接：FROM A RIGHT [OUTER] JOIN B ON A.a = B.b;

全外连接：FROM A FULL [OUTER] JOIN B ON A.a = B.b;

笛卡尔连接(交叉连接):FROM A CROSS JOIN B;

内连接：按照关联条件关联，将两表中都符合关联条件的数据取出

左外连接：按照关联条件关联，左表作为主表，将两表中都符合关联条件的数据以及主表中独有的数据取出

右外连接：按照关联条件关联，右表作为主表，将两表中都符合关联条件的数据以及主表中独有的数据取出

全外连接：按照关联条件关联，将两表中都符合关联关系的数据以及左表中独有的、右表中独有的数据取出

笛卡尔连接：没有关联条件，左表中的每一条数据都与右表中的每一条数据做了一次关联

优化：

- 1) 首先要建立适当的索引。sql 在索引字段不要加函数，保证索引起效。如果是复合索引注意在 sql 的顺序。如果已经存在索引，建议你先重建索引先，因为大数据表的索引维护到了一个阶段就是乱的，一般建议重建。建立好的一般可以获得几十倍的速度提升。
- 2) 最大数据量的表放在最前，最小的表放在最后面。sql 是从最后面开始反向解析的。
- 3) 其次是要把最有效缩小范围的条件放到 sql 末尾去。尤其是主键或者索引字段的条件。
- 4) 保证你 sql 的算法合理性。保证复杂度和空间度的合理性。
- 5) 必要时使用存储过程。提升 30%-40%的速度

6) 建议你分页读取不要一下读完所有的数据。（使用 rownum），一下子数据太多会使得内存不够用的。

59. 聚合函数

SUM () MAX () MIN () COUNT () AVG ()

60. 字符串函数，日期截取，

(1) 连接: CONCAT (STR1, STR2)

(2) 大小写转换函数: UPPER () 将字符全部转换成大写

LOWER () 将全部字符转换成小写

INITCAP () 首字母大写，其他字母小写的

(3) 替换: REPLACE (STR, S1, S2) 将字符串 STR 中的 S1 替换成 S2

(4) 去除函数: TRIM (STR) 去除字符串 STR 两端的空格、去除特殊字符

完整用法: TRIM ([LEADING/TRAILING/BOTH] [S FROM] STR)

LTRIM (STR [S]) 去除字符串左边的空格 (S)

RTRIM (STR [S]) 去除字符串右边的空格 (S)

(5) 截取: SUBSTR (STR, IND, LEN)

(6) 填充: LPAD (STR, N , S) 在字符串 STR 左侧填充 S 字符，使整个长度达到 N

RPAD (STR, N, S) 在字符串 STR 右侧填充 S 字符，使整个长度达到 N

(7) 获取字符串长度: LENGTH (STR)

(8) 获取字节长度: VSIZE (STR)

LENGTHB (STR)

(9) 获取字符位置: INSTR (STR, S, IND, N) 从第 IND 位开始，查找 S 在字符串 STR 中第 N 次出现的位置

日期截取

例如获取当前系统年份: trunc (SYSTADE, 'YYYY')

61. 空值如何处理

--空值可以根据具体要求，运用 NVL(I1, I2)如果 I1 为空值 那么返还 I2 的值或者运用 NVL2 (I1, I2,I3) 函数 如果 I1 为空值那么返还 I3 的值 如果 I1 不为空值则返还 I2 的值

62. 平时用的数据库是什么 oracle 的物理结构是什么

--平时用 ORACLE 数据库，物理结构：在硬盘上的物理组成情况，具体有数据文件、控制文件、重做日志文件、归档日志文件、参数文件、口令文件和警告日志文件

63. Oracle 的数据类型

数字类型

number(L,S) L 表示长度，s 表示精度，最长 38 个长度

Length: 整数类型

pls_integer/binary_integer:只存放整数

旧版数据库:

1. FLOAT(浮点型)小数类型
2. INT:整数类型

字符串类型

1.CHAR(N)定长型 N 表示长度最大为 2000 个字符 不够的系统会自动用空格补齐

2.VARCHAR2()变长字符串，不是空格补齐

3.long: 存放内容比较长

4.ROWID: 存放数据库中伪列 ROWID 的值时使用

日期型

1.DATE

2.TIMESTAMP

布尔类型

Boolean : 布尔类型 只有三种值: TRUE,FALSE,NULL

64. 什么是事物

--"为了完成某项业务（任务、操作），由一系列看得见的 SQL 或看不见的后台进程组成的一系列逻辑工作单元

--最主要的关键词是 COMMIT--事务提交 ROLLBACK--事务回滚"

65. Oracle 里面的存储过程

所谓存储过程，就是一组用于完成特定数据库功能的 SQL 语句集，该 SQL 语句集经过编译后存储在数据库系统中。在使用时候，用户通过指定已经定义的存储过程名字并给出相应的存储过程参数来调用并执行它，从而完成一个或一系列的数据库操作。

Oracle 存储过程包含三部分：过程声明，执行过程部分，存储过程异常。

创建存储过程语法：

不带参数的存储过程：

```
Create [or replace] procedure 存储过程名
```

```
as //声明部分
```

```
;
```

```
begin // 执行部分
```

```
;
```

```
exception//存储过程异常
```

```
;
```

```
end;
```

存储过程完整语法：

```
Create [or replace] procedure 存储过程名 (param1 in type, param2 out type)
```

```
as
```

```
    变量 1 类型 (值范围);
```

```
    变量 2 类型 (值范围);
```

```
begin
```

```
    select count(*) into 变量 1 from 表 A where 列名=param1;
```

```
    if (判断条件) then
```

```
        select 列名 into 变量 2 from 表 A where 列名=param1;
```

```
        dbms_output.Put_line('打印信息');
```

```
    elsif (判断条件) then
```

```
        dbms_output.Put_line('打印信息');
```

```
    else
```

```
        raise 异常名 (NO_DATA_FOUND);
```

```
    end if;
```

```
exception
```

```
    when others then
```

```
        rollback;
```

```
end;
```

参数的几种类型：

in 是参数的默认模式，这种模式就是在程序运行的时候已经具有值，在程序体中值不会改变。

out 模式定义的参数只能在过程体内部赋值，表示该参数可以将某个值传递回调用他的过程

in out 表示高参数可以向该过程中传递值，也可以将某个值传出去。

66. 索引的应用

1.如果对某大表进行筛选时，某列或某几列频繁出现在 **WHERE** 子句中，并且检索出的数据低于总行数的 15%（50%），应考虑在这些列上建立索引。

- 2.如果对某大表进行排序时，某列或某几列频繁出现在 ORDER BY 子句中，应考虑在这些列上建立索引。
- 3.小表不要建立索引。
- 4.对于含有空值的列，如果经常在查询时查询非空值，建议在该列上建立索引；如果经常在查询时查询空值，建议在该列上建立基于函数的索引。
- 5.为了提高表连接的性能，应在连接列上建立索引（建立一般普通的索引即可）
- 6.索引是数据库的一种实体对象，级别类似于表，会占用内存空间，ORACLE 会自动进行索引维护，表和索引可以建立在不同的表空间。
- 7.通过索引可以提升数据的查询速度，但是会相对地降低 DML 语句的操作速度，尤其是插和改的速度，ORACLE 会花费时间在索引维护上，所以说要把握好索引的数量
- 8.对于列基数比较大的列，适合 B-TREE 索引，列基数比较小的列，适合位图索引。
- 9.对于复合索引，至少要引用到索引列中的第一个列才会使用该索引。
- 10.某列可以出现在多个索引中，但相同的某列或某几列无法多次建立索引。 --就是可以有多个组合，但不能有重复组合，不同顺序的相同几列视为不同组合
- 11.索引建立后并不一定会被引用，ORACLE 会分析整个 SQL 后做出最优的执行方式。
- 12.ORACLE 会自动在主键约束和唯一约束列上建立唯一索引。
- 13.对于一般的 B-TREE 索引，通配符出现在搜索词的首位时不会引用索引
- 14.在索引列上使用 <> !=号时，或对空值进行判断时，索引不会生效

67.left join 和 inner join 的区别

left join 连接时返回的是左表中的所有数据和右表中符合关联条件的数据；

inner join 会返回两表符合关联条件的数据；

68.oracle 和 mysql 之间的区别

相同点：1.都是关系型数据库

2.都是目前比较流行的数据库

不同点：

1、体积不同。oracle 是 oracle 的数据库产品，它体积比较庞大，。而 mysql 的体积相对来说比较小，较之 oracle 更容易安装、维护以及管理，操作也简单，

2、容量不同。Oracle 容量无限，根据配置决定；而 MySQL 数据库的最大有效表尺寸通常是由操作系统对文件大小的限制决定的，不是由 MySQL 内部限制决定。

3、平台支持及速度的区别。Oracle 支持大对数平台；而 mysql 支持各种平台，适合 Linux。至于速度，oracle 大部分情况下速度快于 Mysql。

4、性能的区别。Oracle 全面，完整，稳定，；而 mysql 使用 CPU 和内存极少，性能很高，但扩展性较差。

69.使用数据库时所用到的优化

（1）选取最适用的字段属性。

(2) 使用连接来代替子查询。

(3) 使用联合 (UNION) 来代替手动创建的临时表。

(4) 事务。不是所有的数据库操作都可以只用一条或少数几条 SQL 语句就可以完成的。

(5) 锁定表。在执行事务时，锁定表。

(6) 外键，锁定能保证完整性，但是外键也能保持关联性。

(7) 使用索引，索引是提高数据库性能的常用方法。

(8) 优化的查询语句。

70. 介绍一下索引

索引是数据库中对象的一种，它记录了索引列中的数据以及对应的物理位置-ROWID。索引建立时会获取指定列的数据，及其对应的 ROWID，并自动地按照某种规则进行排序，索引主要是为了让表查询效率更高。

71. 索引分类，什么时候用到过位图索引，索引注意什么事项

索引的分类：

按存储形式（即索引中存储的内容不同）：

1) B-TREE 索引(索引列原始数据+ROWID)

2) 位图索引(位图+ROWID)

3) 反向键索引(索引列原始数据的反向存储+ROWID)

4) 基于函数的索引(将索引列原始数据经函数处理后存储+ROWID)

按唯一性（索引列中的数据是否有重复值）：

唯一索引 --索引列中不可能出现重复值

非唯一索引 --索引列中可能出现重复值

按列的个数（索引覆盖的列的个数）

单列索引 --基于一个列建立的索引

复合索引(也称为联合索引) --基于两个或两个以上列建立的索引

列基数小时使用位图索引（例如性别、婚姻状况）

索引会提高查询速度，但是会降低 dml 的操作效率；

小表不要建立索引；

对于含有空值的列，如果经常在查询时查询非空值，建议在该列上建立索引；如果经常在查询时查询空值，建议在该列上建立基于函数的索引；

对于列基数比较大的列，适合 B-TREE 索引，列基数比较小的列，适合位图索引；

对于复合索引，至少要引用到索引列中的第一个列才会使用该索引；

某列可以出现在多个索引中，但相同的某列或某几列无法多次建立索引。 --就是可以有多个组合，但不能有重复组合；

对于一般的 B-TREE 索引，通配符出现在搜索词的首位时不会引用索引；

在索引列上使用 <> != 号时，或对空值进行判断时，索引不会生效；

72.select 查询语句慢怎么办

进行相应的优化，例如：

- 1、应尽量避免在 `where` 子句中使用 `!=` 或 `<>` 操作符，否则将放弃使用索引而进行全表扫描。
- 2、对查询进行优化，应尽量避免全表扫描，首先应考虑在 `where` 及 `order by` 涉及的列上建立索引。
- 3、应尽量避免在 `where` 子句中对字段进行 `null` 值判断，否则将导致引擎放弃使用索引而进行全表扫描
- 4、尽量避免在 `where` 子句中使用 `or` 来连接条件，否则将导致引擎放弃使用索引而进行全表扫描。

73.索引：在什么情况下建索引

- 1、表的主键、外键必须有索引；
- 2、数据量超过 300 的表应该有索引；
- 3、经常与其他表进行连接的表，在连接字段上应该建立索引；
- 4、经常出现在 `Where` 子句中的字段，特别是大表的字段，应该建立索引；
- 5、索引应该建在选择性高的字段上；
- 6、索引应该建在小字段上，对于大的文本字段甚至超长字段，不要建索引；
- 7、复合索引的建立需要进行仔细分析；尽量考虑用单字段索引代替

74.以前用过 ETL 工具嘛？用过多久

用过。ETL 工具的典型代表有：informatica、Datastage、OWB、微软 DTS、Beeload、Kettle

75.学号学科成绩，如何通过 ETL 去配置。

76.ETL 全拼是啥，代表啥意思？

Extract-Transform-Load

ETL 是用来描述将数据从来源端经过抽取（`extract`）、转换（`transform`）、加载（`load`）至目的端的过程。

77.你熟悉 SQL 语句嘛?

SQL 是结构化查询语言(Structured Query Language)，结构化查询语言是一种数据库查询和程序设计语言，用于存取数据以及查询、更新和管理关系数据库系统；sql 语句就是对数据库进行操作的一种语言。

SQL 语句主要有五种分类分别是 DQL 数据查询语言、DML 数据操纵语言、DDL 数据定义语言、TCL 数据控制语言和 TCL 事务控制语言

78.drop,truncate,delete

delete 和 truncate 操作只删除表中数据,而不删除表结构。

但是 drop 语句将删除表的结构、约束、触发器、索引等。

◆属于不同类型的操作

delete 属于 DML,事务提交之后才生效。

而 truncate 和 drop 属于 DDL,操作立即生效、不能回滚。

执行速度: drop > truncate > delete

◆安全性:小心使用 drop 和 truncate,尤其没有备份的时候。具体使用时,想删除部分数据行用 delete,注意带上 where 子句。此外,回滚段要足够大。

◆使用建议:完全删除表使用 drop;想保留表而将所有数据删除,如果和事务无关,使用 truncate,如果和事务有关,或者想触发 trigger,使用 delete。

79.sqlsever 与 oracle 区别

区别

1、操作的平台不同

Oracle 可在所有主流平台上运行, Oracle 数据库采用开放的策略目标,它使得客户可以选择一种最适合他们特定需要的解决方案。客户可以利用很多种第三方应用程序、工具。而 SQL Server 却只能在 Windows 上运行了。

安全性不同

Oracle 的安全认证获得最高认证级别的 ISO 标准认证,而 SQL Server 并没有获得什么安全认证。这方面证明了 Oracle 的安全性是高于 SQL Server 的。

3、性能不同

SQL Server 多用户时性能不佳,而 Oracle 性能最高,保持 windowsNT 下的 TPC-D 和 TPC-C 的世界记录。

4、文体结构不同

Oracle 的文件体系结构为:数据文件 .dbf (真实数据、日志文件 .rdo、控制文件 .ctl、参数文件 .ora。

SQL Server 的文件体系结构为: .mdf (数据字典)、.ndf (数据文件)、.ldf (日志文件)。

5、客户端支持及应用模式不同

SQL Server C/S 结构,只支持 windows 客户,可以用 ADO,DAO,OLEDDB,ODBC 连接。

Oracle 多层次网络计算,支持多种工业标准,可以用 ODBC, JDBC,OCI 等网络客户连接。

80. 关系型数据库

1) 关系数据库

在一个给定的应用领域中，所有实体及实体之间联系的集合构成一个关系数据库。

2) 关系数据库的型与值

关系数据库的型称为关系数据库模式，是对关系数据库的描述，若干域的定义，在这些域上定义的若干关系模式。

关系数据库的值是这些关系模式在某一时刻对应的关系的集合，通常简称为关系数据库。

81. 我有一个表，里面有学号，学科，成绩三个字段，需要你算出成绩总分大于 600 分的学生以及他们的成绩，并且按照成绩从高到低排序，写出语句

```
Create table sc(  
Sno number,  
Cno varchar2(10),  
Score number);  
---按成绩排序  
SELECT DISTINCT SNO,RN FROM(select s.*,sum(score)over(partition by sno) RN FROM SC) WHERE  
RN>600
```

82. Oracle 的导入导出命令

1.将数据库 TEST 中的一个表完全导出 导出到相应路径中

Exp 用户名/密码 @TEST file=路径 full=y

2.将数据库中用户 1 与用户 2 的表导出

exp 用户名/密码@TEST file=路径 owner=(用户 1， 用户 2)

3 将数据库中的表 1 表 2 导出

exp 用户名/密码@TEST file=路径 tables=(表 1,表 2)

1 将 路径 中的数据导入 TEST 数据库中。

imp 用户名/密码@TEST file=路径

如果有的表已经存在，它就会报错，对该表就不进行导入。

在后面加上 ignore=y 就可以了

2 将路径中的 表 1 导入

imp 用户名/密码@TEST file=路径 tables=(表 1)

83.Hive 的分区表, 内表, 外表, 数据倾斜处理, left join, map join, having 的用法, spark executor 的作用

行列转换, 了解存储过程语法块

内部表: 数据存储在 Hive 的数据仓库目录下, 删除表时, 除了删除元数据, 还会删除实际表文件。

外部表: 数据并不存储在 Hive 的数据仓库目录下, 删除表时, 只是删除元数据, 并不删除实际表文件。

分区表: 跟 RDMS 的分区概念类似, 将一张表的数据按照分区规则分成多个目录存储。这样可以通过指定分区来提高查询速度。

数据倾斜: 由于数据分布不均匀, 造成数据大量的集中到一点, 造成的数据热点

(39 题 只找到这么多)

84.SQL 优化储存

- 1.在表中建立索引, 优先考虑 where.group by 使用到的字段。
- 2.查询条件中, 一定不要使用 select *, 因为会返回过多无用的字段会降低查询效率。应该使用具体的字段代替*, 只返回使用到的字段。
- 3.不要在 where 条件中使用左右两边都是%的 like 模糊查询 这样会导致数据库引擎放弃索引进行全表扫描。
- 4.尽量不要使用 in 和 not in, 会造成全表扫描 对于连续的数值, 能用 between 就不要用 in 对于子查询, 可以用 exists 代替
- 5.尽量不要使用 or, 会造成全表扫描。可以用 union 代替 or。
- 6.尽量不要在 where 子句中对字段进行表达式操作, 这样也会造成全盘扫描
- 7.where 条件里尽量不要进行 null 值的判断, null 的判断也会造成全表扫描。应该给字段添加默认值, 对默认值进行判断。
- 8.尽量不要在 where 条件中等号的左侧进行表达式.函数操作, 会导致全表扫描。应该将表达式.函数操作移动到等号右侧。
- 9.程序要尽量避免大事务操作, 提高系统并发能力。
- 10.一个表的索引数最好不要超过 6 个, 如果索引太多的话, 就需要考虑一下那些不常使用到的列上建的索引是否有必要。

85.merge into

语法: merge into 目标表 as 别名
using 数据源 表 视图 子查询 as 别名
on (关联条件)
when matched then - 如果能匹配上
update --对目标表进行更新

```

set A.列名=B.列名 【, A.列名=B.列名】
where 更新条件, 以关联条件成立为前提
delete 对目标表中的旧数据进行删除
where 删除条件 以更新条件成立为前提
when not matched then
insert (A.字段 A.字段……)
values(B.字段 B.字段……) --把数据源中的新数据插入目标表
where 插入条件 --以关联条件不成立为前提

```

86. 查询家庭成员大于三且有两位年龄大于 80 岁的老人的户号

设家庭成员表为 JT, 户号为 HNO, 年龄为 NL

先对户号进行分组计数查询出家庭成员大于三的家庭;

```
SELECT HNO FROM JT GROUP BY HNO HAVING COUNT(1)>3;
```

再用表连接或子查询将上述所得的户号筛选出来, 再对户号进行分组且 where 条件为年龄大于 80, 计数大于 2 的户号即为所得;

```
SELECT A.HNO FROM JT A JOIN (SELECT HNO FROM JT GROUP BY HNO HAVING COUNT(1)>3) B
ON A.HNO=B.HNO WHERE NL>80 GROUP BY A.HNO HAVING COUNT(1)>2;
```

```
SELECT HNO FROM JT WHERE HNO IN (SELECT HNO, FROM JT WHERE AGE>80 GROUP BY HNO
HAVING COUNT(1)>2
) GROUP BY HNO HAVING COUNT(1)>3;
```

87. 代码打印九九乘法口诀

```

BEGIN
  FOR V IN 1..9 LOOP
    FOR I IN 1..V LOOP
      DBMS_OUTPUT.PUT(I||'*'||V||'='||I*V||' ');
    END LOOP;
    DBMS_OUTPUT.NEW_LINE();
  END LOOP;
END;
```

88. 函数练习题一

若 A 为空, 则输出 1, 否则输出本身

NVL(A,1)

若 A 非空，输出 1，否则本身

NVL2(A,1,A)

两边去空格

TRIM()

右去空格

RTRIM()

89.用一条 sql 查询平均分，最高分，最低分，男生最高分，女生最高分
设有一个表 sc 是储存学生的所有信息，其中 ssex 字段储存学生性别信息，score 字段储存学生的成绩信息；

```
SELECT MAX (CASE WHEN SSEX='女' THEN SCORE ELSE 0 END) 女生最高,MAX(CASE WHEN  
SSEX='男' THEN SCORE ELSE 0 END) 男生最高,MAX(SCORE) 最高成绩,MIN(SCORE) 最低成  
绩,AVG (SCORE) 平均成绩 FROM SC;
```

90.自定义函数

它也是一个有名字的 plsql 代码块，它一经创建会保存到数据库中

当数据库启动时，会加载到数据库内存中

它可以在存储过程中使用，也可以在 sql 语句中使用

函数可以有参数，也可以没有参数，但必须有返回值

函数的语法：

create or replace function 函数名[(参数 数据类型,...)]

return 返回值类型

is

声名部分

begin

plsql 代码块;

return 语句;

异常处理代码;

end;

91. 写一个自定义函数，判断两个值的大小

```
CREATE OR REPLACE FUNCTION F1(A NUMBER,B NUMBER)
RETURN VARCHAR2
IS
    V VARCHAR2(30);
BEGIN
    IF A>B THEN
        V:= A||'>'||B;
    ELSIF A=B THEN
        V:= A||'='||B;
    ELSE
        V:= A||'<'||B;
    END IF;
    RETURN V;
END;
```

--调用自定义函数

```
DECLARE
A NUMBER:=&第一个值;
B NUMBER:=&第二个值;
BEGIN
    DBMS_OUTPUT.PUT_LINE(F1(A,B)||';');
END;
```

92. 交集和并集

交集(INTERSECT): 将两个集合中重复的部分选出来，去重并按数据的第一列进行升序排序。

并集 UNION: 将两个的集合的所有数据并在一起，重并按数据的第一列进行升序排序。

UNION ALL: 将两个集合的所有数据并在一起，不进行去重也不排序。

93. 怎么做行列转换

```
SELECT  转换后第一列列名,
        /*CASE WHEN  需转列的值所在列名=需转为列名的值
            THEN  列中的数据 ELSE 0 END,*/
--第一步
```

聚合函数(CASE WHEN 需转列的值所在列名=需转为列名的值
 THEN 列中的数据 ELSE 0 END) AS 转换后的列名,--第二步
 FROM SCORE_1
 GROUP BY 转换后第一列列名;

--pivot 函数：行转列函数：
 语法：pivot(任一聚合函数 for 需转列的值所在列名 in (需转为列名的值));
 执行原理：将 pivot 函数或 unpivot 函数接在查询结果集的后面。相当于对结果集进行处理。
 --unpivot 函数：列转行函数：
 语法：unpivot(新增值所在列的列名 for 新增列转为行后所在列的列名 in (需转为行的列名));
 SELECT * FROM SCORE_2

94. 关系型数据库了解多少，用过哪些

关系数据库是建立在关系数据库模型基础上的数据库，借助于集合代数等概念和方法来处理数据库中的数据。当前主流的关系型数据库有 ORACLE、DB2、SQLServer、SYBASE、INFORMIX、MySQL 等。

95. 讲解你认识的 ETL 和大数据

大数据是指那些数据量特别大、数据类别特别复杂的数据集，这种数据集不能用传统的数据库进行转存、管理和处理，是需要新处理模式才能具有更强大的决策力、洞察发现力和流程优化能力的海量、高增长率和多样化的信息资产

ETL，Extract-Transform-Load 的缩写，是将业务系统的数据经过抽取、清洗转换之后加载到数据仓库的过程。ETL 是数据集成的第一步，也是构建数据仓库最重要的步骤，目的是将企业中的分散、零乱、标准不统一的数据整合到一起，为企业的决策提供分析依据。ETL 一词较常用在数据仓库，但其对象并不限于数据仓库。

96. 索引什么情况下会失效

1、对列进行计算或者使用函数时；2、使用了反向操作或者 link 操作时；3、在 where 语句中使用 or 时，如果有一个列没索引，那么其他列就会失

97. 哪些地方用到了 SQL

98. 去除重复数据

1.select distinct * from tb_name;

```

2.SELECT * FROM TB_NAME
   where rowid in
      (select max(rowid) from TB_NAME group by COL_NAME1,...);
3.select *
   from (select T.*,
      row_number() over(partition by col_name order by 1) rn
      from EMP1 T) t1
   where t1.rn = 1;

```

99. 怎么使用表连接去重

100. 1.地区编号 产品编号 销售量； 查询每个地区销售量前十的产品， 在平时的工作中写过类似的 sql 吗

```

SELECT 销售量 FROM(SELECT 销售量, ROW_NUMBER()OVER(PARTITION BY 地区编号
ORDER BY 销售量 desc) A FROM 表)WHERE A<11;

```

101. A 表目前有 100 万数据，B 表 10 条数据，根据关联键匹配，A 表中的数据与 B 表中的数据关联一一对应，A 表中能关联上的 48 万数据，B 表中能关联上的数据为 5 条。那么问，用 inner join,left join, ringt join, full join 的连接方式，最终查询出多少数据量是多少。

```

inner join : 48 万 条
left join : 100 万条
ringt join : 48 万零五条
full join: 100 万零五条

```

102. 十道银行 oracle 试题，口述

103. sql 语句 都使用过什么

SELECT FROM WHERE GROUP BY HAVING

104. 优化遵循的总体原则是什么

尽量避免在列上进行运算, 这样会导致索引失效.

使用 JOIN 时, 应该用小结果集驱动大结果集.

注意 LIKE 模糊查询的使用, 避免%

仅列出需要查询的字段, 节省内存.

使用批量插入语句节省交互.

不要使用 rand 函数获取多条随机记录.

避免使用 NULL.

不要使用 count(id), 而应该是 count(*)

不要做无谓的排序操作, 而应尽可能在索引中完成排序.

105. rowid 去除重复数据

先查出完全重复的数据, 提取单一的 ROWID, 只留下这单一的 ROWID

106. 视图的作用

1. 简化 (将经常使用的数据定义为视图)
2. 安全 (让用户只能看到和修改自己定义的内容)
3. 在逻辑上具有独立性 (摒弃了原来的表结构)

107. 有哪几种常见的索引

一、按存储形式 (即索引中存储的内容不同)

1. B-TREE 索引 (索引列原始数据+ROWID)
2. 位图索引 (位图+ROWID)
3. 反向键索引 (索引列原始数据的反向存储+ROWID)
4. 基于函数的索引 (将索引列原始数据经函数处理后存储+ROWID)

二、按唯一性 (索引列中的数据是否有重复值)

1. 唯一索引 一索引列中不可能出现重复值

注意点:

- 1) B-TREE 索引可以建立唯一索引, 位图索引不能建立唯一索引
 - 2) 如果在某列上建立了唯一约束或主键约束, ORACLE 会自动在该列上建立一个同名的唯一索引
 2. 非唯一索引 --索引列中可能出现重复值
 - 三、按列的个数 (索引覆盖的列的个数)
 1. 单列索引 --基于一个列建立的索引
 2. 复合索引 (也称为联合索引) --基于两个或两个以上列建立的索引
- 用法: 先用第一个字段, 再写第二个字段

108. 索引需要注意的事项

1. 对大表进行筛选, 某列或者某几列频繁出现在 WHERE 子句中, 并且检索出的数据低于总行数的 15% (50%), 考虑在这些列中建立索引
2. 对某大表进行排序时, 某列或某几列频繁出现在 ORDER BY 子句中, 应考虑在这些列上建立索引
3. 小表不要建立索引
4. 对于含有空值的列, 如果经常在查询时查询非空值, 建议在该列上建立索引; 如果经常在查询时查询空值, 建议在该列上建立基于函数的索引
5. 为了提高表连接的性能, 应在连接列上建立索引 (建立一般普通的索引即可)
6. 索引是数据库的一种实体对象, 级别类似于表, 会占用内存空间, ORACLE 会自动进行索引维护, 表和索引可以建立在不同的表空间。
7. 通过索引可以提升数据的查询速度, 但是会相对地降低 DML 语句的操作速度, 尤其是插和改的速度, ORACLE 会花费时间在索引维护上, 所以说要把握好索引的数量
8. 对于列基数比较大的列, 适合 B-TREE 索引, 列基数比较小的列, 适合位图索引
9. 对于复合索引, 至少要引用到索引列中的第一个列才会使用该索引。
10. 某列可以出现在多个索引中, 但相同的某列或某几列无法多次建立索引。--就是可以有多个组合, 但不能有重复组合, 不同顺序的相同几列视为不同组合
11. 索引建立后并不一定会被引用, ORACLE 会分析整个 SQL 后做出最优的执行方式。
12. ORACLE 会自动在主键约束和唯一约束列上建立唯一索引。
13. 对于一般的 B-TREE 索引, 通配符出现在搜索词的首位时不会引用索引 14. 在索引列上使用 <> !=号时, 或对空值进行判断时, 索引不会生效 15. 需求>设计>程序

109. 刚才说的在索引上使用 like 应该注意什么

使用索引进行模糊查询时, 模糊查询条件不能以通配符开头

110. 什么情况要用外连接

查找多张表时, 有的表中的数据在其他表中不存在

111. 开窗函数了解过吗？都知道些什么？

1. 聚合函数(必有字段名)over(分组:paratition by 排序:order by)
2. 排序函数(必须是空)over(paratition by ,order by(必有))
3. 偏移函数(字段名, 偏移量, 默认值)over(paratition by,order by)

112. 主要使用什么数据库

oracle

113. 左连接与内连接有什么不同？

内连接显示两个表共同的数据，左外连接以左表的数据为基准

114. 给一张表，有三列，一列学生姓名，一列课程名，一个学生可以有多门课程，一列成绩，请问如何查询每个学生自己成绩大于 80 的所有科目？

```
SELECT SNO, COURSE
FROM STUDENT
WHERE GRADE>80
GROUP BY SNO, COURSE
```

115. 索引的使用；

建立索引：

```
CREATE INDEX IND_NAME ON TB_NAME(COL_NAME) ;
```

禁用索引：

```
ALTER INDEX IND_NAME UNUSABLE;
```

重建索引：

```
ALTER INDEX IND_NAME REBUILD;
```

删除索引：

```
DROP INDEX IND_NAME;
```

116. 索引跟主键的区别

1. 应用范畴不同：主键属于索引的一种。在数据库关系图中为表定义主键将自动创建主键索引，主键索引是唯一索引的特定类型。该索引要求主键中的每个值都唯一。当在查询中使用主键索引时，它还允许对数据的快速访问。

种类不同：根据数据库的功能，可以在数据库设计器中创建三种索引：唯一索引、主键索引和聚集索引。而，主键只是其中的一种。

2. 创建方式不同：当创建或更改表时可通过定义 PRIMARY KEY 约束来创建主键。一个表只能有一个 PRIMARY KEY 约束，而且 PRIMARY KEY 约束中的列不能接受空值。由于 PRIMARY KEY 约束确保唯一数据，所以经常用来定义标识列。经常在 WHERE 子句中的列上面创建索引。

注意：

1. 主键一定是唯一性索引，唯一性索引并不一定就是主键。

2. 一个表中可以有多个唯一性索引，但只能有一个主键。

3. 主键列不允许空值，而唯一性索引列允许空值。

117. 创建一个索引，查询、分组、排序中哪个能用到索引

查询, 排序

118. 表连接，左连接会显示多少条数据

左表所有数据

119. sql 优化, exists 与 in

in 是把外表和内表作 hash 连接，而 exists 是对外表作 loop 循环，每次 loop 循环再对内表进行查询。如果查询的两个表大小相当，那么用 in 和 exists 差别不大。如果两个表中一个较小，一个大表，则子查询表大的用 exists，子查询表小的用 in

120. sql 优化，执行计划（主要看什么）

1. sql 如何使用索引 2. 联接查询的执行顺序 3. 查询扫描的数据行数

121. 表关联

内连接：FROM A [INNER] JOIN B ON A.a = B.b --常用

外连接：左外连接：FROM A LEFT [OUTER] JOIN B ON A.a = B.b --常用

右外连接: FROM A RIGHT [OUTER] JOIN B ON A.a = B.b --不常用

全外连接: FROM A FULL [OUTER] JOIN B ON A.a = B.b --偶尔用

笛卡尔连接(交叉连接): FROM A CROSS JOIN B --不常用 笛卡尔积: 比较危险, 会引起数据量暴涨

122. 开窗函数, 三种排数的区别

ROW_NUMBER() 从头到尾直接排

RANK() 出现相同 1 1 3

DENSE_RANK() 出现相同 1 1 2

123. 怎么看这个字段是否用了索引

1. 直接在数据库里查看字段属性
2. 在该字段创建索引查看是否能创建
3. user_ind_columns 根据表名和列名去查

124. 在学校学过 MySQL, 那工作的时候用过 Oracle 吗? 会写存储过程吗?

Oracle 存储过程包含三部分: 过程声明, 执行过程部分, 存储过程异常

1. CREATE OR REPLACE PROCEDURE demo AS/IS
2. 变量 2 DATE;
3. 变量 3 NUMBER;
4. BEGIN
5. --要处理的业务逻辑
6. EXCEPTION --存储过程异常
7. END

125. 重复数据的去重以及重复数据的查找

先利用编号字段分组, COUNT(子段)>1 的就是重复的
再利用 ROWID/DISTINCT 即可去重复

126. Sql 4、5 种语言

DDL(数据定义语言) - Create、Alter、Drop 这些语句自动提交, 无需用 Commit 提交。

DQL(数据查询语言) - Select 查询语句不存在提交问题。

DML(数据操纵语言) - Insert、Update、Delete 这些语句需要 Commit 才能提交。

DTL(事务控制语言) - Commit、Rollback 事务提交与回滚语句。

DCL(数据控制语言) - Grant、Revoke 授予权限与回收权限语句。

127. 为什么会失效

128. 不会或者有歧义的题：15.97.99.127