

# JAVA 面试指南

【数据库部分】



#### 前言

Java 面试指南第 I 版已经推出有一段时间了,感谢大家的支持! 数据库这部分,因为近来工作特别繁忙被迫推辞,向大家致歉! 紧赶慢赶,总算完成一个初步版本,发与大家。

对于数据库部分,这里主要介绍在 Java 面试中常见的 Mysql、Oracle 部分面试,以应用知识为主(对于数据库管理部分涉及较少,有兴趣的还需要自己深入学习)。

关于本部分内容,由于之前面试过程中碰到不多,而且当时未及时总结,现 在是通过回忆的方式来总结的,部分题因为网上总结的很好直接引用了,在此致 谢!(如果引用不当,请指出,及时改正)。

大家看目录的时候,可能感觉到此部分面试题虽然是针对初中级求职人员,但还是有一定难度;笔者认为作为求职者,哪怕你是刚毕业的学生,也应该有一定的数据库基础知识,所以太基础的部分我就没有列出。但对于部分数据库基础差一点的求职者,我这里推荐两本电子书,第一本是《Oracle 教程.pdf》另外一本是《myql 必知必会》,Oracle 教程.pdf 是一本很好的入门学习教程,学习后一些基本的面试题肯定没有问题,我已经放到群共享,以后发放 java 面试指南的时候,也会作为附件发出;另外一本是有实体书的,讲的很好,电子书可能涉及版权,大家可以自行下载,我就不上传了。

本文档只是【Java 面试指南】一部分,请相互结合一起看,由于笔者能力有限,难免有不足之处,欢迎指出;如果有好的建议也可以联系我们dyc gary@126.com,我们将努力改正,不断完善。

Java 面试指南 Core Java 部分可以从群共享下载,或者百度文库下载,全本可以通过下面链接进入。

收获 offer, 挑战高薪, 更多请访问: http://shop113074087.taobao.com

#### 补充说明:

- 1、将你希望看到的内容在 QQ 小窗口中发给我,或者以邮件的形式,我们会整理出在后期的版本中补充,最新的版本会以邮件的形式发给你阅读。
- **2**、数据库部分会根据大家的反馈和笔者收集的资料在后期进行补充,优先发给支付宝下单的同学,期请期待。

## 目录

第一	一部分	数据库基础知识	. 3
	1.1	数据库设计的三大范式	. 3
	1.2	数据库的分页方式	. 3
	1.3	DDL、DML、DCL 的区别	. 4
第二	二部分	Mysql 部分	. 4
	2.1	Mysql 的优越性	. 4
	2.2	mysql 常用的日期函数	. 4
	2.3	如果解决 mysql 的乱码问题	. 5
	2.4	mysql 有哪些常用命令	. 5
	2.5	too many connection 的错误如何解决	. 6
	2.6	mysql 常用的性能优化	. 6
	2.7	mysql 默认连接超接解决	. 8
第三	三部分	Oracle 部分	. 8
	3.1	oracle 左右连接的写法与区别	. 9
	3.2	oracle 函数与过程的区别	11
	3.3	介绍一下 oracle 的游标	11
	3.4	什么是视图	11
	3.5	rownum 与 rowid 区别	12
	3.6	exist 与 in 的区别	12
	3.7	oracle 冷备份与热备份区别	13
	3.8	介绍一下 oracle 索引	13
	3.9	truncate 与 delete 区别	14
	3.10	什么是死锁、如何解决死锁	15
	3.11	union 与 union all 区别	16
	3.12	触发器的种类及作用	17
第四	副部分	数据库笔试部分	17
	4.1	统计姓名重复的人员	17
	4.2	去除表中的重复记录(多字段)	18
	4.3	如何转置(旋转)一张数据库表	18

# 第一部分 数据库基础知识

#### 1.1 数据库设计的三大范式

解析:数据库的三大范式是数据设计的基础,一般开发人员在数据库设计时也都会遵从三大范式,虽会应用也需在面试中能说出来,属于高频率面试题(如果问你数据库设计有几大范式也要会回答),有同学会说数据库不是六大范式吗?这里我们不是去面试 DBA,只要了解基础的三大范式就可以了。下面通过几个小例子给大家加深理解。

#### 第一范式: 保证数据库中的表每一列都是不可以再分的原子列

用户编号	姓名	电话	地址

这里如果需求中想统计这张用户表中的省份信息,市级信息;这里的设计就 违背了数据库设计的第一范式,因为这里的"地址"字段是可以再分的"省、市、区、 街道...."。

#### 第二范式: 在满足第1NF 的基础上, 确保表中的每列都和主键相关

		1 N/ /L / H	1 W/ /I / / // // /
1 1 1 4 4 4 P P	1 1 T 4 K 37 1 T		
	1	1 学生 狮 写	1 字生名称
1 1/1 6/V 5/III 5			
1 グエジスグ四 J			

上面是班级信息表与学生之间的关系,这里大家可能一眼就能看出学生名称与班级编号没有直接关系,而且如果同一个班级有多名学生,这里的班级名称会一直冗余,所以它违背了第 2NF 范式。

#### 第三范式: 在满足第2NF 基础上,属性不能传递依赖于主属性

班级编号   班级名称   学生编号   学生名称
---------------------------

还是通过这张表中来说明,这里的学生名称依赖学生编号,学生名称并不依赖主键"班级编号",所以它违背了第 3NF,应该将两张表拆开。

参考答案:第一范式:保证数据库中的表每一列都是不可以再分的原子列; 第二范式:在满足第1NF的基础上,确保表中的每列都和主键相关;第三范式: 在满足第2NF基础上,属性不能传递依赖于主属性(确保每列都和主键列直接相关,而不是间接相关)。

说明:如果是笔试回答上面绝对可以满分了,如果是面试最好结合例子说明。

## 1.2 数据库的分页方式

解析:这个很简单,mysql 分页使用 limit 关键字,oracle 使用 rownum 关键字,sqlserver 使用关键字 top,这里我就不逐个写例子了,面试的同学自己写一下,笔试中可能遇到。

参考答案: mysql 用 limit 进行分页, oracle 通过 rownum 以嵌套的方式进行分面, sqlserver 通过 top 关键字分页。

#### 1.3 DDL、DML、DCL 的区别

解析:这个是对概念的考查,只需要了解,面试出现不多,笔试可能会出现。

参考答案: DML(Data Manipulation Language)数据操纵语言; 对数据库中的数据进行一些简单操作,如 insert,delete,update,select 等.

DDL(Data Definition Language)数据定义语言:对数据库中的某些对象进行操作,如 create,alter,drop,truncate,(create index)等.

DCL(Data Control Language)数据控制语言,如 commit、 rollback、grant、invoke 等

DML 和 DDL 区别:

1.DML 主要侧重于应用级开发,程序中是可以手动控制事务的开启、提交和回滚的。

2.DDL 主要侧重于服务端管理,不需要考虑事务操作。

# 第二部分 Mysql 部分

推荐大家一本书《mysql 必知必会》,这本书写的挺好的,大家可以从网上下载电子书学习一下,由于 mysql 在实际面试中问到的的确不多,所以我们这里也只是作一些简单介绍,后期如需要会更新补充。

## **2.1** Mysql 的优越性

解析:这是一个给你吹牛的题,只要不吹破就行,现在 mysql 也加入到了 oracle 旗下,将两者对比一下,很容易回答这个问题。

参考答案: mysql 是一个小型开源的关系型数据库管理系统,开发者为瑞典 MySQL AB 公司,先被 SUN 收购、后来 SUN 又被 oracle 收购。Mysql 优点: 1、相对于 oracle、db2、sqlserver 等大型数据库来说,mysql 是一个小型的关系型数据库,安全简单、配置灵活、容易移植、效率高。2、支持标准的 SQL 语句标准,可以很容易的和其它数据库进行转换; 3、可以高效处理千万级别数据,而且成本低,非常适合中小企业使用; 4、丰富网络资源、开发论坛利于学习和员工培训。

## 2.2 mysql 常用的日期函数

解析: 日期函数太多了,这里应该问的是常用的系统日期函数和关键字。

参考答案: mysql 常用的系统日期函数有 now(),sysdate(),curdate()函数 Current timestamp 关键字。

## 2.3 如果解决 mysql 的乱码问题

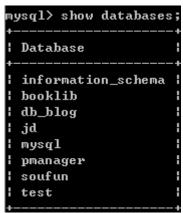
解析:这个题出现频率不高,主要结合应用考察,一般首先要分析说明一下乱码产生的原因是在应用端还是在服务端,然后再解决。

参考答案: 1.在安装数据库的过程中将默认的拉丁文-->GBK。

- 2.在创建数据库时设置选择 GBK 或者 gb2312。
- 3.Mysql 安装目录下的 my.ini 文件,将 "default-character-set=xxxxx" 中的 xxxxx 改成 GBK 或者 gb2312。
- 4.Mysql 安装目录下的\data\databasename(数据库名)\db.opt 文件打开 default-character-set=gbk default-collation=gbk\_chinese\_ci; 如果上面不是 gbk 和 gbk\_chinese\_ci 则改成支持中文的 GBK 或者 gb2312。
- 5. 进入 Mysql 的 dos 命令下:进入某数据库后 show full columns from tablename;查看数据类型,如果不是支持中文的类型则执行 alter table tablename convert to character set gbk 。
- 6. 在创建数据库时(用命令创建时)create database databasename CHARACTER SET gbk (gb2312);

## 2.4 mysql 有哪些常用命令

解析:这里的命令一般是通过 shell 脚本或者命令窗口中执行的指令,这里以 mysql.exe 命令窗口为例说明。



像一些数据库连接、授权、DML操作、数据库备份都可以通过命令完成,这 里写一些常用的。

参考答案: 1、连接数据库 mysql -h 主机地址 -u 用户名 -p 用户密码 2、快捷操作 show databses: 列出所有的数据库; use databasesname:打开 此数据库; show tables:查看当前数据库下面的所有表,然后可以对表进行 DML

操作; describe table\_name: 查看某表下的字段和类型的定义。 3 、在 use databasename 状态下导入表 SOURCE d:/mysgl.sql;

退出命令 quit; 或者 exit

## 2.5 too many connection 的错误如何解决

解析:这是一个实际应用问题,这类的问题无论面试频率是否高,都会在现实工作中遇到,所以建议必须掌握;这里报 too many connection 说明我们的连接数不够用,连接占用时间太长没有释放;解决方法一般是增大连接数,优化查询语句提高查询效率,使用一些缓存策略,对数据库进行集群部署,最不济时可以考虑写文件。

参考答案:造成 too many connection 的直接原因是连接数被占满了,解决这种异常可以通过下面几种方案解决。

- 一、扩大连接数: mysql 的最大连接数默认是 100, 这个数值对于并发连接很多的数据库应用是远远不够的,可以通过编辑编辑 my.ini 修改max\_connections=1000 增大连接数。
- 二、连接数过多产生的另外一个原因是因为连接没有及时释放或者没释放,可以从以下方面考虑。1、优化相关的 SQL 语句,提高执行效率,及时释放连接;2、养成良好的开发习惯,及时的关闭连接;适当使用数据源连接池,减少创建物理连接造成的额外开销;3、适当的使用缓存策略,减少与数据库的直接交互:4、如果有条件可以进行数据集群,增大数据库的吞吐量
  - 三、寻找替换方案,比如可以写文件,可以使用内存相关的数据库nosql。

## 2.6 mysql 常用的性能优化

解析:这个经常会被问到,出现频率较高,至少能回答一个三五条,下面列
Challenge high salary Challenge OFFER

#### 出来一些常用到的优化方法,大家可以选择了解。

- 参考答案: 1.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。
- 2. 应尽量避免在 where 子句中使用!=或<>操作符,否则将引擎放弃使用索引而进行全表扫描。
- 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.应尽量避免在 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,
- 5. %xx%和%xx 查询也将导致全表扫描,若要提高效率,可以考虑全文检索或者xx%模糊查询。
- 6.in 和 not in 也要慎用,否则会导致全表扫描,对于连续的数值,能用 between 就不要用 in 了
- 7.如果在 where 子句中使用参数,也会导致全表扫描。因为 SQL 只有在运行时才会解析局部变量,但优化程序不能将访问计划的选择推迟到运行时;它必须在编译时进行选择。
- 8.应尽量避免在 where 子句中对字段进行表达式操作,这将导致引擎放弃使用索引而进行全表扫描。
- 9.应尽量避免在 where 子句中对字段进行函数操作,这将导致引擎放弃使用索引而进行全表扫描。
- 10. 不要在 where 子句中的 "=" 左边进行函数、算术运算或其他表达式运算, 否则系统将可能无法正确使用索引。
- 11.在使用索引字段作为条件时,如果该索引是复合索引,那么必须使用到该索引中的第一个字段作为条件时才能保证系统使用该索引,否则该索引将不会被使用,并且应尽可能的让字段顺序与索引顺序相一致。
- 12.索引并不是越多越好,索引固然可以提高相应的 select 的效率,但同时也降低了 insert 及 update 的效率,因为 insert 或 update 时有可能会重建索引,所以怎样建索引需要慎重考虑,视具体情况而定。一个表的索引数最好不要超过 6 个,若太多则应考虑一些不常使用到的列上建的索引是否有必要。
- 13.尽量使用数字型字段,若只含数值信息的字段尽量不要设计为字符型, 这会降低查询和连接的性能,并会增加存储开销。这是因为引擎在处理查询和 连接时会逐个比较字符串中每一个字符,而对于数字型而言只需要比较一次就 够了。
- 14.尽可能的使用 varchar/nvarchar 代替 char/nchar ,因为首先变长字段存储空间小,可以节省存储空间,其次对于查询来说,在一个相对较小的字段内搜索效率显然要高些。
- 15.任何地方都不要使用 select \* from t ,用具体的字段列表代替 "\*",不要返回用不到的任何字段。
- 16. 尽量使用表变量来代替临时表。如果表变量包含大量数据,请注意索引非常有限(只有主键索引)。
- 17.在新建临时表时,如果一次性插入数据量很大,那么可以使用 select into 代替 create table,避免造成大量 log ,以提高速度;如果数据量不大,为了缓和系统表的资源,应先 create table,然后 insert。

18.尽量避免向客户端返回大数据量,若数据量过大,应该考虑相应需求是否合理。

## 2.7mysql 默认连接超时解决

解析:这个问题在实际开发中经常用到,笔者之前在一个 OA 项目中也曾经碰到过,mysql 默认有一个时间,超过这个时间,登录就会报超时的异常;这是一个结合实际应用的面试问题。

参考答案: 1、JDBC 连接: connection url 中加参数: autoReconnect=true jdbc.url=jdbc:mysql://ipaddress:3306/database?autoReconnect=true&autoReconnectForPools=true(话说只能在mysql5之前的版本有效, mysql5之后的版本我没有亲自测试)。

我没有亲自测试)。
2、HIBERNATE 连接:需要增加如下属性
<property name="connection.autoReconnect">true</property>
<property name="connection.autoReconnectForPools">true</property>
<property name="connection.is-connection-validation-required">true</property>
3、使用c3p0 连接池:(我们的解决方式)
<property name="hibernate.c3p0.acquire\_increment">1</property>
<property name="hibernate.c3p0.idle\_test\_period">0</property>
<property name="hibernate.c3p0.timeout">0</property>
<property name="hibernate.c3p0.validate">true</property>
4、通过命令修改'wait\_timeout'的值(需要重启 mysql 服务器):
use mysqlname; -- 在 mysql 库下。
SHOW GLOBAL VARIABLES LIKE 'wait\_timeout'; -- 查看当前值。
SET GLOBAL wait timeout = 1814400; -- 修改(这里为21 天单位秒)。

说明: 也可以通过修改 mysql 配置文件进行修改(windows-my.ini;linux-my.cnf)

# 第三部分 Oracle 部分

推荐大家一本书《Oracle 教程.pdf》,电子书我已经上传到群共享,大家可以下载学习;这是一本很好的 oracle 入门书,可以帮你达到中级水平。对于之前购买《Java 面试指南》的朋友,我会用邮件再发一下。

对于 oracle 部分及笔试相关部分,下面的用例皆以 orace 中 scott 用户下面的 EMP(雇员表),DEPT(部门表),SALGRADE(工资等级表),BONUS(奖金表)为 基础表说明,scott 用户默认是锁定的,安装 oracle 的同学可以通过 DBA 权限的 用户把 scott 用户解锁(Oracle 学习见群共享资料)。

1、雇员表(EMP)

	雇员表 (EMP)									
No.	字段	类型	描述							
1	EMPNO	NUMBER(4)	表示雇员编号,是唯一编号							
2	ENAME	VARCHAR2(10)	表示雇员姓名							
3	JOB	VARCHAR2(9)	表示工作职位							
4	MGR	NUMBER(4)	表示一个雇员的领导编号							
5	HIREDATE	DATE	表示雇佣日期							
6	SAL	NUMBER(7,2)	表示月薪,工资							
7	COMM	NUMBER(7,2)	表示奖金,或者称为佣金							
8	DEPTNO	NUMBER(2)	部门编号							

#### 2、部门表(DEPT)

	部门表 (dept)								
No.	字段 类型 描述								
1	DEPTNO	NUMBER(2)	部门编号,是唯一编号						
2	DNAME	VARCHAR2(14)	部门名称						
3	LOC	VARCHAR2(13)	部门位置						

#### 3、工资等级(SALGRADE)

	工资等级表(SALGRADE)								
No.	字段 类型 描述								
1	GRADE	NUMBER	等级名称						
2	LOSAL	NUMBER	此等级的最低工资						
3	HISAL	NUMBER	此等级的最高工资						

#### 4、奖金表(BONUS)

	奖金表 (BONUS)								
No.	字段	类型	描述						
1	ENAME	VARCHAR2(10)	雇员姓名						
2	JOB	VARCHAR2(9)	雇员工作						
3	SAL	NUMBER	雇员工资						
4	COMM	NUMBER	雇员奖金 (佣金)						

## 3.1 oracle 左右连接的写法与区别

解析:这是笔者刚毕业在北京去东软面试的面试题目,很简单很基础的一道题目,出现的频率很高。

Oracle 的连接有多种方式:内连接、左外连接、右外连接、交叉连接、不等连接、自连接等,不过不用担心,在实际开发和面试中用不到这么多,这里只需要掌握**内连接和外连接(左外、右外、全外连接)**。

内连接: 等值连接, 最常用的连接方式, 不多说。

Exe:select \* from emp e ,dept d where e.deptno = d.deptno

左外连接: 用 left join ..on..或者用(+)表示

Challenge high salary Challenge OFFER

Exe1:select \* from emp e ,dept d where e.deptno = d.deptno(+);
Exe2:select \* from emp e left join dept d on e.deptno = d.deptno;

		EMPNO _	ENAME	JOB	MGR	HIREDATE		SAL _	сомм	DEPTNO	DEPTNO _	DNAME	LOC _
Þ	1	7934	MILLER	CLERK	7782	1982/1/23	•	1300.00		10	10	ACCOUNTING	NEW YORK
	2	7839	KING	PRESIDENT		1981/11/17	•	5000.00		10	10	ACCOUNTING	NEW YORK
	3	7782	CLARK	MANAGER	7839	1981/6/9	•	2450.00		10	10	ACCOUNTING	NEW YORK
	4	7902	FORD	ANALYST	7566	1981/12/3	•	3000.00		20	20	RESEARCH	DALLAS
	5	7876	ADAMS	CLERK	7788	1987/5/23	•	1100.00		20	20	RESEARCH	DALLAS
	6	7788	SCOTT	ANALYST	7566	1987/4/19	•	3000.00		20	20	RESEARCH	DALLAS
	7	7566	JONES	MANAGER	7839	1981/4/2	•	2975.00		20	20	RESEARCH	DALLAS
	8	7369	SMITH	CLERK	7902	1980/12/17	•	800.00		20	20	RESEARCH	DALLAS
	9	7900	JAMES	CLERK	7698	1981/12/3	•	950.00		30	30	SALES	CHICAGO
	10	7844	TURNER	SALESMAN	7698	1981/9/8	•	1500.00	0.00	30	30	SALES	CHICAGO
	11	7698	BLAKE	MANAGER	7839	1981/5/1	•	2850.00		30	30	SALES	CHICAGO
	12	7654	MARTIN	SALESMAN	7698	1981/9/28	•	1250.00	1400.00	30	30	SALES	CHICAGO
	13	7521	WARD	SALESMAN	7698	1981/2/22	•	1250.00	500.00	30	30	SALES	CHICAGO
	14	7499	ALLEN	SALESMAN		1981/2/20	+	1600.00	300.00	30	30	SALES	CHICAGO
	15	7937	TEST	CLERK	7782	1982/1/23	•	1111.00		50			

右外连接: 用 left join ..on..或者用(+)表示

Exel:select \* from emp e ,dept d where e.deptno(+) = d.deptno;

Exe2:select \* from emp e right join dept d on e.deptno = d.deptno;

		EMPNO _	ENAME	JOB	MGR	HIREDATE		SAL _	COMM	DEPTNO	DEPTNO _	DNAME	LOC _
•	1	7369	SMITH	CLERK	7902	1980/12/17	•	800.00		20	20	RESEARCH	DALLAS
	2	7499	ALLEN	SALESMAN	7698	1981/2/20	-	1600.00	300.00	30	30	SALES	CHICAGO
	3	7521	WARD	SALESMAN	7698	1981/2/22	•	1250.00	500.00	30	30	SALES	CHICAGO
	4	7566	JONES	MANAGER	7839	1981/4/2	•	2975.00		20	20	RESEARCH	DALLAS
	5	7654	MARTIN	SALESMAN	7698	1981/9/28	•	1250.00	1400.00	30	30	SALES	CHICAGO
	6	7698	BLAKE	MANAGER	7839	1981/5/1	•	2850.00		30	30	SALES	CHICAGO
	7	7782	CLARK	MANAGER	7839	1981/6/9	•	2450.00		10	10	ACCOUNTING	NEW YORK
	8	7788	SCOTT	ANALYST	7566	1987/4/19	•	3000.00		20	20	RESEARCH	DALLAS
	9	7839	KING	PRESIDENT		1981/11/17	•	5000.00		10	10	ACCOUNTING	NEW YORK
	10	7844	TURNER	SALESMAN	7698	1981/9/8	•	1500.00	0.00	30	30	SALES	CHICAGO
	11	7876	ADAMS	CLERK	7788	1987/5/23	•	1100.00		20	20	RESEARCH	DALLAS
	12	7900	JAMES	CLERK	7698	1981/12/3	-	950.00		30	30	SALES	CHICAGO
	13	7902	FORD	ANALYST	7566	1981/12/3	•	3000.00		20	20	RESEARCH	DALLAS
	14	7934	MILLER	CLERK	7782	1982/1/23	•	1300.00		10	10	ACCOUNTING	<b>NEW YORK</b>
	15						•				40	OPERATIONS	BOSTON

全外连接: 用 full outer join ..on..或 full join..on.. 不能用 (+) 号表示

Exe1:select \* from emp e full outer join dept d on e.deptno =
d.deptno;

Exe2:select \* from emp e full join dept d on e.deptno = d.deptno;

	EMPNO _	ENAME _	JOB _	MGR	HIREDATE		SAL _	COMM	DEPTNO _	DEPTNO _	DNAME	LOC _
1	7937	TEST	CLERK	7782	1982/1/23	٠	1111.00		50			
2	7369	SMITH	CLERK	7902	1980/12/17	+	800.00		20	20	RESEARCH	DALLAS
3	7499	ALLEN	SALESMAN	7698	1981/2/20	•	1600.00	300.00	30	30	SALES	CHICAGO
4	7521	WARD	SALESMAN	7698	1981/2/22	-	1250.00	500.00	30	30	SALES	CHICAGO
5	7566	JONES	MANAGER	7839	1981/4/2	•	2975.00		20	20	RESEARCH	DALLAS
6	7654	MARTIN	SALESMAN	7698	1981/9/28	•	1250.00	1400.00	30	30	SALES	CHICAGO
7	7698	BLAKE	MANAGER	7839	1981/5/1	•	2850.00		30	30	SALES	CHICAGO
8	7782	CLARK	MANAGER	7839	1981/6/9	•	2450.00		10	10	ACCOUNTING	<b>NEW YORK</b>
9	7788	SCOTT	ANALYST	7566	1987/4/19	•	3000.00		20	20	RESEARCH	DALLAS
10	7839	KING	PRESIDENT		1981/11/17	•	5000.00		10	10	ACCOUNTING	<b>NEW YORK</b>
11	7844	TURNER	SALESMAN	7698	1981/9/8	•	1500.00	0.00	30	30	SALES	CHICAGO
12	7876	ADAMS	CLERK	7788	1987/5/23	•	1100.00		20	20	RESEARCH	DALLAS
13	7900	JAMES	CLERK	7698	1981/12/3	•	950.00		30	30	SALES	CHICAGO
14	7902	FORD	ANALYST	7566	1981/12/3	•	3000.00		20	20	RESEARCH	DALLAS
15	7934	MILLER	CLERK	7782	1982/1/23	•	1300.00		10	10	ACCOUNTING	<b>NEW YORK</b>
16						•				40	OPERATIONS	BOSTON

参考答案: 左连接和右连接都是属于外连接,又称左外连接、右外连接, 左连接可以用 left join..on..或者(+),右连接可以 right join..on..或者用(+) 两种 写法; 左连接显示的结果是左侧的表不加限制(左侧表中的数据都显示),右 连接是右侧的表不加限制(右侧表中的数据都显示)。

补充:如果用 left(right)..join..on 的方式很容易分清左右两表,如果用 (+)则以 where 后面的条件的顺序为准。

扩展:全外连接如果用左外和右外的方式如何写?

#### 3.2 oracle 函数与过程的区别

解析: 这是一个很容易面试的问题,而且频率很高。

参考答案: 1、函数与过程都是数据库中的对象,函数声明用 function, 过程声明用 procedure。

- 2、函数通常封装的是一些算法,必须有返回值;过程通是用来处理一系列具有业务意义的算法或者语句块,可以没有返回值,也可以返回单个值或者返回多个值。
- 3、函数必须在 DML 语句中调用,但过程不能在 DML 语句中直接调用,它必须单独的调用,可以作为语句块调试。

## 3.3 介绍一下 oracle 的游标

解析:这是一个概念题,不会问很深,在理解的基础上进行记忆,下面介绍一些游标的类别、游标状态。

参考答案: 游标定义就是从表中检索出结果集,从中每次指向一条记录进行交互的机制(或者可以理解为存放 select 集的一块内存区域)。Oracle 中的游标可以分为静态游标和 REF 游标(我比较喜欢用),静态游标又分为显示游标(必须声明定义)和隐式游标;游标的常用属性%rowcount 记录数,%found影响到了记录,%notfound 没有影响到记录,%isopen 是否打开。

扩展:游标的使用最好也能说一些,这里东西太多,建议看一下推荐的教程。

#### 3.4 什么是视图

解析:这是一个概念题,视图是一个虚拟表,视图并不在数据库中存储数据值,数据库中只在数据字典中存储对视图的定义。视图在实际开发中用到的很多,一般情况下问的不会很深(DBA除外),但作为实际工作来说也是必须掌握的。

参考答案: 视图是数据库的一个对象,封装了比较复杂查询的虚拟表。 Oracle 中的视图主要分为关系视图、内嵌视图、对象视图、物化视图。

关系视图: 最常用的视图, 创建之后可以通过工具或数据字典来查看视图的相关信息。

内嵌视图:内嵌的意义在于,视图定义嵌入在复杂查询语句中,其角色与普通数据表或关系视图相同。内嵌视图不必使用 create view 命令进行创建,因此,在数据字典中也无法获得相应信息。

对象视图: Oracle 中的对象数据实际仍然以关系数据的形式存储, 利用 Oracle 内置视图 user\_views 可以获得对象视图相关信息,具有封装、继承等对象 的牲,可以构造复杂的对象类型来封装复杂的多表查询。

物化视图:常用于数据库的容灾,不是传统意义上虚拟视图,是实体化视图,和表一样可以存储数据、查询数据。主备数据库数据同步通过物化视图实现,主备数据库通过 data link 连接,在主备数据库物化视图进行数据复制。视图的优点:1.对数据库的访问,因为视图可以有选择性的选取数据库里

的一部分。 2.用户通过简单的查询可以从复杂查询中得到结果。 3.维护数据的独立性,试图可从多个表检索数据。 4.对于相同的数据可产生不同的视图。 5.视图可以屏蔽表的内部结构,提高安全。

说明,这里的概念很不容易理解,最好动手去操作一下,上面最常用的是关系视图和物化视图;最不容易理解的是内嵌视图,当前最少用到的是对象视图(以后可能会越来越常用)。

#### 3.5 rownum 与 rowid 区别

参考答案: 1. rowid 和 rownum 都是虚列 2. rowid 是物理地址,用于定位 oracle 中具体数据的物理存储位置 3. rownum 则是 sql 的输出结果排序,根据查询的语句默认从 1 开始递增。4。 rowid 常用来做一些重复数据过滤之类的操作,而且 rownum 常用在分页语句中。

## 3.6 exist 与 in 的区别

解析: 这是一个频率很高的面试题,如果面试数据库部分,就很容易被问到。 首先来看一下 exist 和 in 的等价写法(来源网络) 比如 Select \* from T1 where x in ( select y from T2 ) 等价于 select \* from t1, ( select distinct y from t2 ) t2 where t1.x = t2.y; select \* from t1 where exists ( select null from t2 where y = x ) 等价于

for x in ( select \* from t1 )

loop

if ( exists ( select null from t2 where y = x.x )

then

OUTPUT THE RECORD

end if

end loop

从上面的执行可以看出,对于 in 的执行是外表与内表的 hash 连接,而对于 exists 则相当于 loop 循环,再对内表进行查询。

IN 适合于外表大而内表小的情况,EXISTS 适合于外表小而内表大的情况。

In 确定给定的值是否与子查询或列表中的值相匹配。

Exists 指定一个子查询, 检测行的存在。

exits 相当于存在量词:表示集合存在,也就是集合不为空只作用一个集合.例如 exist P表示 P不空时为真; not exist P表示 p为空时为真 in 表示一个标量和一元关系的关系。例如: s in P表示当 s 与 P中的某个值相等时为真; s not in P表示 s 与 P中的每一个值都不相等时为真

参考答案: 1、exists 适用于主表小而从表大的情况,而 in 相反用在主表大 从表小或者固定集合的情况。

- 2、in 确定给定的值是否与子查询或列表中的值相匹配。
- 3、exists 指定一个子查询, 检测行的存在, 是一个存在判断, 如果后面的查

询中有结果,则 exists 为真,否则为假。

4、not in 和 not exists 如果查询语句使用了 not in 那么内外表都进行全表扫描,没有用到索引;而 not extsts 的子查询依然能用到表上的索引;所以无论那个表大,用 not exists 都比 not in 要快。

#### 3.7 oracle 冷备份与热备份区别

解析:冷备份就是在数据库关闭的情况下进行备份,可以直接 COPY 文件,简单易操作;热备份是在数据库运行的情况下进行备份,必须是 archive log 模式下,这里不多作总结,网上有一位朋友总结的相当好,这里直接拿过来分享一下。

参考答案: 冷备份发生在数据库已经正常关闭的情况下,将需要备份的 文件进行拷备处理(最简单的方式就是整个库进行 COPY); 热备份是在数据库 运行的情况下,采用归档方式备份数据,必须是在 archive log 模式下进行。

#### 冷备份的优点>>>

- 1. 备份方法速度快,容易归档
- 2. 容易恢复到某个时间点上
- 3. 与数据库归档的模式相结合可以使数据库很好地恢复。
- 4. 维护量少,安全度高。

#### 冷备份缺点>>>

- 1. 单独使用时,只能提供到"某一时间点上"的恢复。
- 2. 在实施备份的全过程中,数据库必须要作备份而不能作其它工作。也就是说,在冷备份过程中,数据库必须是关闭状态。
  - 3. 若磁盘空间有限,只能拷贝到磁带等其它外部存储设备,速度会很慢。
  - 4. 不能按表或按用户恢复。

#### 热备的优缺点>>>

- 1. 可在表空间或数据文件级备份,备份时间短。
- 2. 备份时数据库仍可使用。
- 3. 可达到秒级恢复。
- 4. 可对几乎所有数据库实体作恢复。
- 5. 恢复是快速的,在大多数情况下在数据库仍工作时恢复。

#### 热备份的缺点>>>

- 1. 不能出错,否则后果严重。
- 2. 若热备份不成功,所得结果不可用于时间点的恢复。
- 3. 因难于维护,所以要特别仔细小心,不允许"以失败而告终"。

#### 3.8 介绍一下 oracle 索引

解析:索引在 oracle 中也是一个经常被面试问到的地方,并且索引包括的东西很多,这里主要通过索引定义、索引的作用、索引的类型、索引的创建原则、索引的创建讨论一下(建议结合教材实际操作一下,若不然禁不起面试人官的追问)。

参考答案: 1、索引概念:索引是类似于书本目录,用于加速数据存取的数据对象,是一种供服务器在表中快速查找一个行的数据库结构(简单说就是可以更快的查询表中数据)。

- 2、索引作用: 加速度数据库存取数据; 合理使用可以减少磁盘 IO 的读取次数,提高查询效率; 可以对列进行唯一性约束; 实现表与表之间的参照性完整性。
- 3、索引类型:按存储分可以分为B\*索引和位图索引(B\*树索引的存储结构类似书的索引结构,有分支和叶两种类型的存储数据块,分支块相当于书的大目录,叶块相当于索引到的具体的书页。位图索引存储主要用于节省空间,减少 oracle 对数据块的访问。它采用位图偏移方式来与表的行 ID 号对应,采用位图索引一般是重复值太多的表字段)。

按功能分类: 唯一索引(不会有两行相同记录); 非唯一索引(不对索引列进行唯一限制); 分区索引(索引分散地存在多个不同的表空间,可以提高查询效率); 复合索引(对多个字段进行联合建立索引); 正向索引(创建索引时不必指定对其排序而使用默认的顺序); 反向索引(该索引同样保持列按顺序排列,但是颠倒已索引的每列的字节)等。

- 4、索引的创建原则:索引字段建议建立NOT NULL 约束;经常与其他表进行连接的表,在连接字段上应该建立索引;经常出现在Where 子句中的字段且过滤性很强的,特别是大表的字段,应该建立索引;可选择性高的关键字,应该建立索引;可选择性低的关键字,但数据的值分布差异很大时,选择性数据比较少时仍然可以利用索引提高效率;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。
  - 5、索引创建语法: (顺序不同,索引不同) 单索引: create index 索引名 on 表名(列名); 复合索引: create index 索引名 on 表名(列名,列名...);
  - 6、索引的优缺点:

优点: 1). 创建唯一性索引,保证数据库表中每一行数据的唯一性 2). 大大加快数据的检索速度,这也是创建索引的最主要的原因 3). 加速表和表 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。缺点: 1). 索引创建在表上,不能创建在视图上 2). 创建索引和维护索 引要耗费时间,这种时间随着数据量的增加而增加 3). 索引需要占物理空间, 除了数据表占数据空间之外,每一个索引还要占一定的物理空间,如果要建立 聚簇索引,那么需要的空间就会更大 4). 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

#### 3.9 truncate 与 delete 区别

解析:该问题在面试的时候经常被问到,要能够说得明白。

参考答案: 1、truncate 是属于 DDL,delete 是属于 DML 操作,语法结构为 delete from tablename,truncate table tablename 两者都可以删除表中数据,但不 会删除表结构。

2、delete 是逐条删除,每条删除的记录都会记录日志,删除效率低,事后

可以根据日志进行数据恢复; truncate 是按页删除,日志只记录页释放,效率高,但事后无法根据日志进行恢复。

- 3、delete from 在行级上删除,可以更加细致控制要删除的数据,只需要在where 后面加上限制条件,但truncate 不可以。
- 4、delete 使用行级锁执行删除,将锁定表中的各行以便删除; truncate 始终锁定表和页而非是行。
- 5、delete 执行删除后,会被放到 rllback segment 中,需要提交整条后生产,如果绑定了 tigger,会执行触发器;而 truncate 不需要事务提交,不会对 tigger 进行触发 。

扩展:与 drop 作一下对比。

#### 3.10 什么是死锁、如何解决死锁

解析:死锁概念并不默生,在 java 多线程中也经常看到这个词,解决死锁首先要定位产生死锁的原因,这个面试中出现的概率并不是特别高,了解一下便可。

参考答案: 死锁产生的原因,主要是加了锁而又忘记提交事务或者忘记数据回滚造成的。Oracle 中的死锁一般是行级死锁和表级死锁,如果是行级死锁,被锁定的行将不能再操作;如果是表级死锁,那么会看到应用端一直在等待。死锁的解决方法(万能方法,重启数据库):1、定位产生死锁的位置2、杀死相应的进程。(下面这些表的操作往往是要管理员权限)

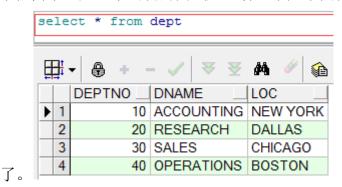
1). 查找出被锁的表

select b.owner,b.object\_name,a.session\_id,a.locked\_mode from v\$locked\_object a,dba\_objects b where b.object\_id = a.object\_id; select b.username,b.sid,b.serial#,logon\_time from v\$locked\_object a,v\$session b where a.session\_id = b.sid order by b.logon\_time; --这里会列出 SID 2). 杀进程中的会话 alter system kill session "sid,serial#";

说明:如果不是DBA,可能记不住定位死锁的一些操作命令或者要查哪些表,但至少能说出操作步聚。

#### 3.11 union 与 union all 区别

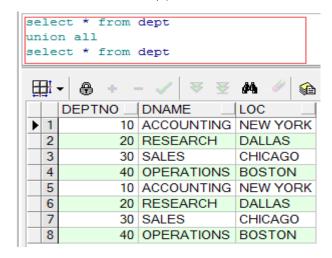
解析: 这是一个很简单, 又经常出现的面试题, 看一下几个图很容易就理解



(图一)

un	select * from dept union select * from dept									
₫	<u>]</u> ;		+ -	- /   ¥ ¥ A /   @						
		DEPT	NO	DNAME LOC _						
Þ	1		10	ACCOUNTING NEW YORK						
	2		20	RESEARCH DALLAS						
	3		30	SALES CHICAGO						
	4		40	OPERATIONS BOSTON						

(图二)



(图三)

上面 "图一"是 dept 原表中的数据, "图二"是 union 两张表后的数据, 和图一一样, 说明过滤重复记录; "图三"将 dept 的数据显示两遍, 说明有重复数据。

参考答案: union 、unilon all 作用在两张表一样的表上,都是对两表查询的结果求并集合。union 查询出来的结果,会过滤掉重复行,并且按照默认规则排序。union all 查询出来的结果,不会进行重复过滤,也不会进行排序。如果确认两表中没有重复数据,建议用 union all 代替 union,union all 效率要高

此。

扩展: 自己动手总结一下 intersect 、minus 的用法。

## 3.12 触发器的种类及作用

解析: 触发器是一个很重要的概念,在笔者参与的面试过程中并没有被直接问到过,但触发器面试频率依然很高,而且很难局部了解,建议结合教程系统学习一下。

参考答案: 触发器概念, 它是一个特殊的存储过程, 它是基于事件由 oracle 系统自动触发, 常用来加强数据的完事性约束和业务规则验证。

触发器的种类: DML 触发器、替代触发器、系统触发器。

1.DML 触发器

ORACLE 可以在 DML 语句进行触发,可以在 DML 操作前或操作后进行触发即为 BEFORE 触发器和 AFTER 触发器,并且可以对每个行或语句操作上进行触发又为语句级触发器和行级触发器。

2. 替代触发器

由于在ORACLE 里,不能直接对由两个以上的表建立的视图进行操作。所以给出了替代触发器。

3. 系统触发器

系统触发器在发生如数据启动或关闭等系统事件时触发,包括数据库服务器的启动或关闭、用户登录与退出、数据库服务错误等。

# 第四部分 数据库笔试部分

对于这部分,大家把我推荐的《oracle 教程》看完看懂就没有什么问题了,如果没有看到的,可以通过群共享文件下载。下面只是把出现频率高,大家容易忘记的单独总结。

## 4.1 统计姓名重复的人员

```
题目: 统计下面 t_test 表中,人员 name 重复出现两次(包括两次)以上的名单。create table t_test ( --//这里的id为主键 id number(2) , name varchar(2) ) insert into aa values(1, 'A') ; insert into aa values(2, 'B') ; insert into aa values(3, 'C') ; insert into aa values(4, 'B') ;
```

```
insert into aa values(5,'D');
insert into aa values(6,'E');
insert into aa values(7,'F');
insert into aa values(8,'A');
insert into aa values(9,'B');
insert into aa values(10,'A');
insert into aa values(11,'D');
insert into aa values(12,'G');
insert into aa values(13,'H');
解析: 这是一个很简单的题,考的是 having 关键字,不多说,必须会的。

参考答案: select name from t_test group by name having count(*)>=2;
```

#### 4.2 去除表中的重复记录(多字段)

```
题目:删除多字段重复的表中的重复记录,同样的记录只显示一条。
```

```
create table TEMP
(
   ID NUMBER,
   NAME VARCHAR2(15),
   TEL VARCHAR2(11)
);
insert into TEMP (ID, NAME, TEL)values (1, 'tom', '111');
insert into TEMP (ID, NAME, TEL)values (2, 'tom', '111');
insert into TEMP (ID, NAME, TEL)values (3, 'jack', '121');
insert into TEMP (ID, NAME, TEL)values (4, 'ketty', '131');
insert into TEMP (ID, NAME, TEL)values (5, 'jack', '121');
commit;
```

解析:此题与上题基本一致,唯一区别就是字段多些,此题是笔者第一份工作的面试题,大家必须会的。

```
参考答案: 第一种方式: delete from temp where id not in(select min(id) from temp group by name, tel);
第二种方式: delete from temp
where id in (select al.id
    from temp al, temp a2
    where al.id > a2.id
    and al.name = a2.name
    and al.tel = a2.tel)
```

扩展: 如果只展示不重复的记录如何显示, 自己动手写一下。

## 4.3 如何转置(旋转)一张数据库表

题目:有如下二维表:

姓名	课程编号	课程	成绩
S1	C1	数据结构	91
S1	C2	操作系统	81
S1	C3	微积分	71
S2	C1	数据结构	92
S2	C2	操作系统	82
S2	C3	微积分	72
S3	C1	数据结构	93
S3	C2	操作系统	83
S3	C3	微积分	73

#### 要转置得到如下二维表:

姓名	数据结构	操作系统	微积分
S1	91	81	71
S2	92	82	72
S3	93	83	73

#### 要求:

1. 只使用一条语句

create table student(

- 2. select 姓名,数据结构,操作系统,微积分 from 表 order by 姓名;
- 3. 可以使用自己熟悉的任何一种数据库的 SQL 语法实现。 解析: 这是一道很精典的题,下面是我建的表结构,大家可以自己动手写下。

```
name varchar2(10),
cno char(2),
cource varchar2(10),
score number(2)
)
insert into student values('S1','C1','数据结构','91');
insert into student values('S1','C2','操作系统','81');
insert into student values('S1','C3','微积分','71');
insert into student values('S2','C1','数据结构','92');
insert into student values('S2','C2','操作系统','82');
insert into student values('S2','C3','微积分','72');
insert into student values('S3','C1','数据结构','93');
insert into student values('S3','C2','操作系统','83');
insert into student values('S3','C3','微积分','73');
```

```
参考答案: select s11. name "姓名", s11. score "数据结构", s12. score "操作系统", s13. score "微积分" from (select name, score from student where cno = 'C1') s11, (select name, score from student where cno = 'C2') s12, (select name, score from student where cno = 'C3') s13 where s11. name = s12. name and s12. name = s13. name
```