Oracle笔试题库附参考答案

1． 下列不属于ORACLE 的逻辑结构的是（C ）

1. 区
2. 段
3. 数据 件
4. 表空间

2.下 哪个 户不是ORACLE 缺省安装后就存在的 户(A)

A . SYSDBA

B. SYSTEM

C. SCOTT

D. SYS

3 下 哪个操作会导致 户连接到 ORACLE 数据库，但不能创建表(A)

1. 授予了CONNECT 的 ，但没有授予RESOURCE 的
2. 没有授予 户系统管理员的
3. 数据库实例没有启动
4. 数据库监听没有启动

1.（ 函数通常 来计算累计排名，移动平均数和报表聚合。

A .汇总

B. 分析

C 分组、

D 单

1. 带有（B ）字句的SELECT 语句可以在表的 或多 放置排他锁。

A . FOR INSERT

B. FOR UPDATE

C. FOR DELETE

D. FOR REFRESH

1. 在Oracle中，你以SYSDBA 登录，CUSTOMER 表位于Mary 户 案中，下 哪条语句为数据库中的所有 户创建CUSTOMER 表的同义词（B ）。
    1. CREATE PUBLIC SYNONYM cust ON mary.customer;
    2. CREATE PUBLIC SYNONYM cust FOR mary.customer;
    3. CREATE SYNONYM cust ON mary.customer FOR PUBLIC;
    4. 不能创建CUSTOMER 的公 同义词。
    5.

7.在Oracle中，当FETCH 语句从游标获得数据时，下 叙述正确的是 C ）。

1. 游标打开
2. 游标关闭
3. 当前记录的数据加载到变量中
4. 创建变量保存当前记录的数据

8.在Oracle中，下 关于函数描述正确的是（AD ）。

1. SYSDATE 函数返回Oracle服务器的 期和时间
2. ROUND 数字函数按四舍五 原则返回指定 进制数最靠近的整数
3. ADD_MONTHS 期函数返回指定两个 份天数的和
4. SUBSTR 函数从字符串指定的位置返回指定长度的 串

9.阅读下 的PL/SQL 程序块：

```
BEGIN
INSERT INTO employee(salary,last_name,first_name)
VALUES(35000,’ Wang’,’Fred’);
SAVEPOINT save_a;
INSERT INTO employee(salary,last_name,first_name)
VALUES(40000,’ Woo’,’David’);
SAVEPOINT save_b;
DELETE FROM employee WHERE dept_no=10;
SAVEPOINT save_c;
INSERT INTO employee(salary,last_name,first_name)
VALUES(25000,’ Lee’,’Bert’);
ROLLBACK TO SAVEPOINT save_c;
INSERT INTO employee(salary,last_name,first_name)
```

VALUES(32000,’ Chung’,’Mike’);

ROLLBACK TO SAVEPOINT save_b;

COMMIT;

END;

运　上　的程序，哪两个更改永久保存到数据库 ©D ）。

1. DELETE FROM employee WHERE dept_no=10;
2. INSERT INTO employee(salary,last_name,first_name)
    1. i. VALUES(32000,’ Chung’,’Mike’);
    2. INSERT INTO employee(salary,last_name,first_name)
        1. i. VALUES(35000,’ Wang’,’Fred’);
        2. INSERT INTO employee(salary,last_name,first_name)
            1. i. VALUES(40000,’ Woo’,’David’);

10.在Oracle中，表VENDOR　包含以下列：

VENDOR_ID NUMBER Primary Key

NAME VARCHAR2(30)

LOCATION_ID NUMBER

ORDER_DT DATE

ORDER_AMOUNT NUMBER(8,2)

下　对表VENDOR　运　分组函数的　句合法的是 ©C ）。

1. FROM MAX(order_dt)
2. SELECT SUM(order_dt)
3. SELECT SUM(order_amount)
4. WHERE MAX(order_dt) = order_d

11.在Oracle中，表EMP 包含以下列：

……

NAME VARCHAR2(20)

ADDR VARCHAR2(60)

……

要以NAME’s address is ADDR格式返回数据，以下SQL 语句正确的是（B ）。

+ 的两边做To_number()

1. SELECT NAME + ’ ’ ’ s address is ‘ + ADDR FROM EMP;
2. SELECT NAME || ’ ’ ’ s address is ‘ || ADDR FROM EMP;
3. SELECT NAME + ’ \’ s address is ‘ + ADDR FROM EMP;
4. SELECT NAME || ’ \’ s address is ‘ || ADDR FROM EMP;

12.在Oracle中，以下不属于集合操作符的是（B ）。

1. UNION
2. SUM
3. MINUS
4. INTERSECT

13.在Oracle中，表分区　式（C ）建议分区数是2的幂（2、4、8等），以获得最平均的数据发布。

1. 范围分区
2. 列表分区
3. 散列分区
4. 复合分区

14.在Oracle中，关于锁,下列描述不正确的是（D ）。

1. 锁　于在　户之间控制对数据的并发访问
2. 可以将锁归类为　级锁和表级锁
3. insert update、delete语句　动获得　级锁
4. 同　时间只能有　个　户锁定　个特定的表

15 .在Oracle中，关于表分区下列描述不正确的是（）。

1. 分区允许对选定的分区执　维护操作，　其他分区对于　户仍然可
2. 不可以对包含LONG 或LONG RAW　列的表进　分区
3. 不可以对包含任何LOB 列的表进　分区
4. 如果分区键包含DATE 数据类型的列，则必须使 TO_DATE　函数完整的指定年份

16:

在Oracl中，哪种表分区维护操作可以将　分区表转换为分区表的分区（）。

1. 添加分区
2. 结合分区
3. 交换表分区
4. 合并分区

17:

在Oracle中，使　带有（）　句的SELECT　命令可以在表的　　或多　上放置排他锁。

1. FOR INSERT OF
2. FOR REFRESH
3. FOR UPDATE OF
4. FOR LOCK

18:

在Oracle中，序列venseq 使　下　的语句创建：

CREATE SEQUENCE venseq INCREMENT BY 1 START WITH 10

MAXVALUE 100 MINVALUE 10 CYCLE CACHE 5;

下　对序列venseq 修改的语句，错误的是（A ）。

1. ALTER SEQUENCE venseq START WITH 1;
2. ALTER SEQUENCE venseq MAXVALUE 90;
3. ALTER SEQUENCE venseq NOMINVALUE;
4. ALTER SEQUENCE venseq NOCACHE;

19:

在Oracle中，使　以下语句创建视图：

CREATE OR REPLACE VIEW pen_view

AS SELECT * FROM order_master

WHERE ostatus = ‘p’ WITH CHECK OPTION CONSTRAINT penv;

当　户试图执　以下语句:

UPDATE pen_view SET ostatus = ‘d’ WHERE ostatus=’p’;

下列描述正确的是（B ）。

1. Oracle将ostatus修改为d，不提　任何错误
2. Oracle不执　更新操作，并返回错误信息
3. Oracle将ostatus修改为d，同时返回错误信息
4. Oracle不执　更新操作，也不提　任何错误

20:

在Oracle中，在以下情形建　位图索引可以获得较好的性能（C ）。

1. 在列值唯　的列上建
2. 在列值相差很　的列上建
3. 在列值频繁重复的列上建
4. 在　于保存位图的列上建

21:

在Oracle中，通过命令（C ）可以释放锁。

1. INSERT
2. DELETE
3. ROLLBACK
4. UNLOCK

22

在Oracle中，下　关于簇的描述不正确的是（B ）。

1. 簇是　组表，这些表拥有公共列，并且经常　起使
2. 与每个表及其索引单独存储相　，簇能够提　INSERT 语句的性能
3. 由于簇表中的所有　都使　相同的列作为公共主键，因此这些列仅为所有表存储　次，从　节省了存储空间
4. 经常更新的列不应选作簇键

23:

在Oracle中，抽象数据类型address_ty定义如下：

CREATE TYPE address_ty AS OBJECT

(street_no number(3),

street_name varchar2(20),

city varchar2(20),

state varchar2(20))

/

表vend_mast 定义如下：

CREATE TABLE vend_mast

(vencode varchar2(5), venname varchar2(15),

venadd address_ty, tel_no number(10));

下列语句能正确执　的是（）。

1. SELECT city FROM vend_mast WHERE venname=’ Joe’ ;
2. UPDATE vend_mast SET address_ty.street_no=10

WHERE venname=’ john’ ;

1. DELETE FROM vend_mast a WHERE a.venadd.vencode=’ v100’ ; 需要启别名才能访问
2. SELECT a.venadd.street_name FROM vend_mast a

WHERE street_no=11;

24:

在Oracle中，（）数据字典表可以 来查看抽象数据类型的实际结构。

1. USER_TYPE_ATTRS
2. USER_TABLES
3. USER_TYPE_TABS
4. USER_TAB_COLUMNS

25:

在Oracle中，（B ）操作符允许引 现有的 对象。

1. CAST
2. REF
3. MULTISET
4. MAP

26:

在Oracle中，关于PL/SQL 下列描述正确的是（C ）。

1. PL/SQL 代表Power Language/SQL
2. PL/SQL 不 持 向对象编程
3. PL/SQL 块包括声明部分、可执 部分和异常处理部分
4. PL/SQL 提供的四种内置数据类型是character,integer,float,boolean

27:

在Oracle中，阅读下列PL/SQL 块：

DECLARE

v_lower NUMBER:=2;

v_upper NUMBER:=100;

v_count NUMBER:=1;

BEGIN

I = 2; I <= 2; i++

FOR i IN v_lower..v_lowerLOOP

INSERT INTO test(results)

VALUES (v_count);

v_count := v_count +1;

ENDLOOP;

END;

请问FOR LOOP 循环体执 了（A ）次。

1. 1
2. 2
3. 98
4. 100
5. 235:

28:

在Oracle中，（）程序包中提供了所有预定义的异常的定义。

1. DBMS_STANDARD
2. DBMS_STD
3. DBMS_SQL
4. DBMS_TYPES

29:

在Oracle中，关于 程序的描述不正确的是（C ）。

程序 就是存储过程和函数

1. 程序是已命名的PL/SQL 块，可带参数并可在需要时随时调
2. 程序可以具有声明部分、可执 部分和异常处理部分
3. 程序参数的模式只有IN和OUT 两种模式
4. 程序可分为过程和函数两种类型

30:

在Oracle中，当从SQL 表达式调 函数时，下列描述不正确的是（AC ）。

1. 从SELECT 语句调 的函数均不能修改数据库表
2. 函数可以带有IN、OUT 等模式的参数
3. 函数的返回值必须是数据库类型，不得使 PL/SQL 类型
4. 形式参数必须使 数据库类型，不得使 PL/SQL 类型

31:

在Oracle中，关于触发器的描述正确的是（D ）。

1. 触发器可以删除，但不能禁
2. 触发器只能 于表
3. 触发器可以分为 级和语句级两种
4. 触发器是 个对关联表发出insert update、delete或select … for update语句时触发的存储过程

32:

在Oracle中，你需要创建索引提 薪 审查的性能，该审查要对员 薪 提 12个百分点后进 分析处理，下 哪个create index命令能解决此问题（A ）。

1. CREATE INDEX my_idx_1 ON employee(salary*1.12);
2. CREATE UNIQUE INDEX my_idx_1 ON employee(salary);
3. CREATE BITMAP INDEX my_idx_1 ON employee(salary);
4. CREATE INDEX my_idx_1 ON employee(salary) REVERSE;

33:

在Oracle中，执 下 的语句：

SELECT ceil(-97.342),

floor(-97.342), -123.01 -124

round(-97.342),

trunc(-97.342)

FROM dual;

哪个函数的返回值不等于-97（B ）。

1. ceil()
2. floor()
3. round(0)
4. trunc()

34:

在Oracle中， 以下SQL 命令创建了 个序列：

CREATE SEQUENCE my_seq

START WITH 394

INCREMENT BY 12

NOMINVALUE

NOMAXVALUE

NOCYCLE

NOCACHE;

户执 包含my_seq.NEXTVAL 的SQL 语句三次，然后执 包含(seq建好后的第 次my_seq.NEXTVAL 值不增加还是394)

my_seq.CURRVAL 的SQL 语句四次，请问序列my_seq 的当前值是（B ）。

1. 406
2. 418
3. 430
4. 442
5. 242:

35:

在Oracle中，下列哪种标量类型不能保存到数据库表中（B ）。

1. CHAR
2. RAW
3. DATE
4. BOOLEAN

36:

在Oracle中，不属于游标属性的是（C ）。

1. %NOTFOUND
2. %FOUND
3. %ISCLOSE
4. %ISOPEN

37:

在Oracle中，在SQL 提 符下调 PL/SQL 过程的 法是（ABC ）。

1. 使 CALL 语句
2. 使 EXECUTE 语句
3. 使 RUN 语句
4. 直接使 过程名

38:

在Oracle中，　下列语句定义了　个过程：

```
CREATE OR REPLACE PROCEDURE test(a IN VARCHAR2,

b IN OUT NUMBER,

c OUT VARCHAR2) IS

BEGIN

……

END;

/
```

假定使　的变量都已定义，下　对过程test的调　语法正确的是（C ）。

1. test(‘String1’,50,v_str2)
2. test(v_str1,v_num1,’String2’)
3. test(‘String1’,v_num1,v_str2)
4. test(v_str1,20,v_str2)

39:

在Oracle中，关于程序包的描述不正确的是（B ）。

1. 程序包是　种数据库对象，它是对相关PL/SQL 类型、　程序、游标、异常、变量和常量的封装
2. 程序包中私有对象是通过PRIVATE　关键字来标识的 程序包体可以包括没有在程序包说明中列出的对象，这些是私有对象，程序包的　户不能使
3. PL/SQL 允许两个或多个打包的　程序具有同　名称，只要　程序接受的参数数据类型不同
4. 程序包具有模块化、信息隐藏、新增功能及性能更佳等优点

40:

在Oracle中，　户（A ）拥有所有内置的程序包。

1. SYS
2. SYSTEM
3. PUBLIC
4. DEFAULT

41:

在Oracle中，DBMS_LOB　程序包的（）过程　于删除指定的BLOB 或CLOB 。

1. delete
2. erase
3. remove
4. empty_lob

42

在Oracle中，关于触发器的描述正确的是（D ）。（选择　项）

1. 触发器可以删除，但不能禁
2. 触发器只能　于表
3. 触发器可以分为　级和语句级
4. 触发器使　个对关联表发出select、insert、update或delete语句时触发的存储过程

43:

在Oracle中，通过命令（CD ）可以释放锁。（选择两项）

1. INSERT
2. DELETE
3. ROLLBACK
4. COMMIT

44:

在Oracle中，事务中使　下列SQL 语句不会引起锁定（A ）。（选择　项）

1. SELECT
2. INSERT
3. UPDATE
4. DELETE

45:

在Oracle中，当需要使　显式游标更新或删除游标中的　时，声明游标时指定的SELECT　语句必须带有（C ）　句。（选择　项）

1. WHERE CURRENT OF
2. INTO
3. FOR UPDATE
4. ORDER BY

46:

在Windows 操作系统中，Oracle的（A ）服务器监听并接受来　客户端应　程序的连接请求。（选择　项）

1. OracleHOME_NAMETNSListener
2. OracleServiceSID
3. OracleHOME_NAMEAgent
4. OracleHOME_NAMEHTTPServer

47:

在Oracle中，有　个名为seq的序列对象，以下语句能返回序列值但不会引起序列值增加的是（C ）。（选择　项）

1. select seq.ROWNUM from dual;
2. select seq.ROWIDfrom dual;
3. select seq.CURRVALfrom dual;
4. select seq.NEXTVALfrom dual;

48:

ORACLE　中，执　语句：SELECT address1||’,’||address2||’,’||address2　”Address” FROM employees;将会返回（B ）列。

1. 0
2. 1
3. 2
4. 3

49:

在Oracle中，INSTEAD OF　触发器主要　于（D ）。（选择　项）

1. 表
2. 表和视图
3. 基于单个表的视图
4. 基于多个表的视图

50:

Oracle数据库中，下　（C ）可以作为有效的列名。

1. Column
2. 123_NUM
3. NUM_#123
4. #NUM123

51:

在Oracle中，以下　具可以逻辑备份数据库对象或整个数据库（B ）。（选择　项）

1. SQL*Plus
2. 导出实　程序
3. 导　实　程序
4. SQL*Loader

52:

Oracle数据库中，通过（B ）访问能够以最快的　式访问表中的　　。

1. 主键
2. Rowid
3. 唯　索引
4. 整表扫描

53:

在Oracle中，使　下列的语句产　序列：

CREATE SEQUENCE id　　；

Oracle服务器会预开辟内存并维持的序列值有（C ）个。（选择　项）

1. 0
2. 10
3. 20
4. 100

54:

Oracle数据库中，以下（C ）命令可以删除整个表中的数据，并且　法回滚。

1. drop
2. delete
3. truncate
4. cascade

55:

Oralce数据库中，以下（A ）函数可以针对任意数据类型进　操作。

1. TO_CHAR
2. LOWER
3. MAX
4. CEIL

56:

在Oracle中，语句（B ）将ORDER_DATE　　期值’2006年3　18　’显　为‘2006年1　1　’。（选择　项）

1. SELECT ROUND(order_date,’ day’)FROM inventory
2. SELECT ROUND(order_date,’ YEAR’)FROM inventory
3. SELECT ROUND(order_date,’ month’)FROM inventory
4. SELECT ROUND(to_char(order_date,’ yyyy’))FROM inventory

57:

在Oracle中，以下是STUDENTS　表的结构：

SID NUMBER NOT NULL　，Primary Key

SNAME VARCHAR2(30)

COURSE_ID VARCHAR2(10) NOT NULL

MARKS NUMBER

你需要查询参加了课程ID为C10 的考试，并且分数排在前10名的学 ，以下（D ）语句能够实现此功能。

A. SELECT SID,marks,ROWNUM “Rank”

FORM students

WHERE ROWNUM<=10 AND course_id=’ C10 ′

ORDER BY marks DESC;

B.SELECT SID,marks,ROWNUM”Rank”

FORM students

WHERE ROWNUM<=10 AND course_id=’ C10 ′

ORDER BY marks;

C.

SELECT SID,marks,ROWNUM”Rank”

FORM (SELECT SID ,marks

FORM students

WHERE ROWNUM<=10 AND course_id=’ C10 ′

ORDER BY marks DESC;

D.

SELECT SID,marks,ROWNUM”Rank”

FORM (SELECT SID ,marks

FORM students

WHERE course_id=’ C10 ′ ORDER BY marks DESC)

WHERE ROWNUM<=10;

58:

在Oracle中， 下列语句定义 个过程:

CREATE PROCEDURE proc(value1 in varchar2,value2 out number,value3 in out varchar2)

is

begin

……

end;

假定使 的变量都已定义,下 对过程proc的调 语法正确的是().选择 项）

1. proc(‘tt’ ,10,v3)
2. proc(‘tt’ ,v2,v3)
3. proc(v1,v2,v3)
4. proc(‘tt’ ,10,’ dd’)

59:

在Oracle中,使 以下语句创建视图;

create or replace view myView

as select * from orders

where status=’ p’;

假定Orders表中包含10条status=’的记录,当 户试图执 以下语句;

update myView set status=’ o’ where status=’ p’;

下列正确的是()

1. Oracle不执 更新操作,并返回错误信息
2. Oracle成功执 更新操作,再次查看视图时返回0 记录
3. Oracle成功执 更新操作,再次查看视图时返回10 记录
4. Oracle执 更新操作,但提 错误信息

60:

在Oracle中， 于PL/SQL 程序输出调试信息的内置程序包是D

1. DBMS_STANDARD
2. DBMS_ALERT
3. DBMS_LOB
4. DBMS_OUTPUT

61:

在Oracle中， 下列语句定义了 个过程:

CREATE PROCEDURE proc(value1 IN VARCHAR2,value2 OUT NUMBER,value3 IN OUT VARCHAR2)

IS

BEGIN

……

END;

假定使　的变量都已定义，下　对过程proc的调　语法正确的是(B)。

1. proc(‘tt’,10,v3)
2. proc(‘tt’,v2,v3)
3. proc(v1,v2,v3)
4. proc(‘tt’,10,’dd’)

62:

在Oracle中，使　以下语句创建视图：C

CREATE OR REPLACE VIEW MyView

AS SELECT * FROM orders

Where status=’p’;

假定orders表中包含10条status=’的记录，当　户试图执　以下语句：

UPDATE MyView SET status=’o’ WHERE status=’p’;

下列描述正确的是()。

1. Oracle不执　更新操作，并返回错误信息
2. Oracle成功执　更新操作，再次查看视图时返回0　记录
3. Oracle成功执　更新操作，再次查看视图时返回10　记录
4. Oracle执　更新操作，但提　错误信息

63:

在Oracle中,语句()将ORDER_DATE　　期值’2006年3　18　’显　为’2006年1　1　’(选择两项)

1. select round(order_date,’day’)from inventory
2. select round(order_date,’year’)from inventory
3. select round(order_date,’month’)from inventory
4. select round(to_char(order_date,’yyyy’))from inventory

64:

在Oracle中,使　下列语句产　序列:

CREATE SEQUENCE id;

Oracle服务器会预开辟内存并维持的序列值有()个(选择　项)

1. 0
2. 10
3. 20
4. 100
5. 292:

65:

在Oracle中,以下　具可以逻辑备份数据库对象或整个数据库().

1. sql*plus
2. 导出实　程序
3. 导　实　程序
4. sql*loader

66:

在Oracle中,INSTEAD OF 触发器主要　于()

1. 表
2. 表和视　图
3. 基于单个表的视图
4. 基于多个表的视图

67:

在Oracle中，　于显　PL/SQL 程序输出调试信息的内置程序包是()。

1. DBMS_STANDARD
2. DBMS_ALERT
3. DBMS_LOB
4. DBMS_OUTPUT

68:

在Oracle中，有　个名为seq的序列对象，以下语句能返回序列值但不会引起序列值增加的是()。

1. select seq.ROWNUM form dual;
2. select seq.ROWID form dual;
3. select seq.CURRVAL form dual;
4. select seq.NEXTVAL form dual;

69:

在Oracle中，事务中使　下列SQL 语句不会引起锁定()。

1. SELECT
2. INSERT
3. UPDATE
4. DELETE

70.

在Oracle中，通过命令()可以释放锁。

1. INSERT
2. DELETE
3. ROLLBACK
4. COMMIT

71

在Oracle中，关于触发器的描述正确的是()。

1. 触发器可以删除，但不能禁
2. 触发器只能　于表
3. 触发器可以分为　级和语句级
4. 触发器是　个对关联表发出select、insert、update或delete语句是触发的存储过程

72:

ORACLE　中，执　语句： SELECT address1||', '||address2||''||address2　“Adress” FROM emp将会返回（）列。（选择　项）

1. 0
2. 1
3. 2
4. 3

73:

在Oracle中，PL/SQL 块中定义了　个带参数的游标：

CURSOR emp_cursor(dnum NUMBER) IS

SELECT sal,comm FORM emp WHERE deptno=dnum;

那么正确打开此游标的语句是(AD)。

1. OPEN emp_cursor(20);
2. OPEN emp_cursor FOR 20;
3. OPEN emp_cursor USING 20;
4. FOR rmp_rec IN emp_cursor(20) LOOP … ENDLOOP;

74:

Oracle数据库中，下　（）可以作为有效的列名。（选择　项）

1. 75: Column
2. 123_NUM
3. NUM_#123
4. #NUM123

Oracle数据库中，通过（B）访问能够以最快的　式访问表中的　　。（选择　项）

1. 主键
2. Rowid
3. 唯　索引
4. 整表扫描

76:

Oracle数据库中，以下()函数可以针对任意数据类型进　操作。

1. TO_CHAR
2. LOWER
3. MAX
4. CEIL

77:

Oracle数据库中，以下（）命令可以删除整个表中的数据，并且　法回滚。（选择　项）

1. drop
2. delete
3. truncate
4. cascade

78:

在Oracle中，下列(B)语句不能　于控制游标。

1. Open
2. Create
3. Fetch
4. Close

79:

在Oracle中有表”cd”它包含属性”cdcode,”” category和”cdname”; 要查询category取值为”CLASSIC”或”ROCK”的　，应采　语句(AD)。

1. SELECT * FROM cd WHERE category IN (‘CLASEIC’,’ROCK’);
2. SELECT * FROM cd WHERE category BETWEEN ‘CLASSIC’ AND ‘ROCK’;
3. SELECT * FROM cd WHERE category=’CLASSIC’ AND category=’ROCK’;
4. SELECT * FROM cd WHERE category=’CLASSIC’ OR category=’ROCK’;

80:

在Oracle中，在执　SQL 语句时，你不　　使　Update 命令将所有的ID值设置成了11111，那么使　(C)命令可以取消这　操作。

1. EXIT
2. COMMIT
3. ROLLBACK
4. UNDO

81

在Oracle中，数据库中的触发器是　个对关联表发出insert、update或（A）语句时触发的存储过程。（选择　项）

A　　delete

B　　drop

C　create

D　truncate

82

在Oracle中，使　了如下的语句创建　户TOM ，则对于该　户　　，以下说法错误的是（D ）。（选择　项）CREATE USER TOM IDENTIFIED BY TOMSYS

A　该　户的　令为TOMSYS

B　TOM默认为表空间为SYSTEM

C　TOM的临时表空间为TEMP

D　使ORANT UPDATE　　命令可以修改TOM 的　令

83

在Oracle中，下述（A ）命令会使挂起的事务完成。（选择　项）。（选择　项）

A　COMMIT

B　DELETE

C　UPDATE

D　SELECT

e)　INSERT

84

游标变量的类型是(B)

A　隐式游标

B　显　游标

C　RE游标

D　循环游标

85

在　归档　式下操作的数据库禁　了（A ）。（选择　项）

A　归档　志。

B　联机　志。

C　志写　程序。

D　志　件。

86

由于软硬件问题导致的读写数据库　件失败，属于（D ）故障。（选择　项）

A　实例

B　语句

C　户进程

D　介质

87

(C)参数　于确定是否要导　整个导出　件

A　constranints

B　tables

C　full

D　file

88

在oracle程序中处理语句时发　的逻辑错误导致(C)故障

A　实例

B　介质

C　语句

D　户进程

89

以下哪种备份　式需要在完全关闭数据库后进　。

A　归档　志模式。

B　归档　志模式。

C　使　导出实　程序进　逻辑备份。

D　以上都不对。

90

(B) 式的导出会从指定的表中导出所有的数据。（选择　项）

A　分区

B　表

C　全部数据库

D　表空间

91

使　（B）命令可查看在创建触发器时发　的编译错误

A　View errors

B　Show errors

C　Display errors

D　Check errors

92

(A)包　于显　pl/sq块和存储过程中的调试信息

A　Dbms_output

B　Dbms_standard

C　Dbms_Input

D　Dbms_session

93

(A)触发器允许触发操作的语句访问　的列值

A　级 :old :new

B　语句级

C　模式

D　数据库级

94

要审计　户执　的create,drop和alte等ddl语句，应创建(E 触发器

A　级

B　语句级

C　Instead of

D　模式

e) 数据库

95

Oracle内置程序包由(A) 户所有

A　sys

B　system

C　scott

D　Pub lic

96

(D)程序包　于读写操作系统　本　件

A　Dbms_output

B　Dbms_lob图 ，电影)

C　Dbms_random

D　Utl_file

97

以下不属于命令的pl/sq块的是( )

A　程序包

B　过程。

C　游标。

D　函数

98

执　特定任务的　程序是(AB)

A　函数

B　过程。

C　程序包

D　游标。

99

　程序的(A)模式参数可以在调　　程序时指定　个常量

A　　in

B　　out

C　　In out

D　　inout

100

下　关于主事务处理和　主事务处理的说法错误的是(D)

A　　主事务处理结果的变化不倚赖于主事务处理的状态

B　　主事务处理提交或回退时，不影响主事务处理的结果

C　　主事务处理提交，对主事务处理是不可见的

D　　主事务处理还可以启动其他　主事务处理

```
Create   proc p1
Begin
Insert …..
P2();
Update ….
Commit;
End;

Create proc p2
Begin
Atuo……
Inusert….
Rollback;
End;
```

```
   JDBC 查询学　成绩单,把主要代码写出来（考试概率极　).
Connection cn = null;
PreparedStatement pstmt =null;
Resultset rs = null;
try
{
 Class.forname(driveClassName);
 cn =  DriverManager.getConnection(url,username,password);
 pstmt = cn.prepareStatement(“select  score.* from score ,student  “ +
   “where score.stuId = student.id and student.name = ?”);
 pstmt.setString(1,studentName);
 Resultset rs = pstmt.executeQuery();
 while(rs.next())
 {
  system.out.println(rs.getInt(“subject”)  +  “    ” + rs.getFloat(“score”) );
 }
}catch(Exception e){e.printStackTrace();}
finally
{
 if(rs != null) try{ rs.close() }catch(exception e){}
 if(pstmt != null) try{pstmt.close()}catch(exception e){}
 if(cn != null) try{ cn.close() }catch(exception e){}
}
```

1. Oracle跟SQL Server 2005的区别?
宏观上：
1.最　的区别在于平台，oracle可以运　在不同的平台上，sql server只能运　在windows 平台上，由于windows 平台的稳定性和安全性影响了sql server的稳定性和安全性
2. oracle使　的脚本语　为PL-SQL ，　sql server使　的脚本为T-SQL
微观上： 从数据类型,数据库的结构等等回答

2.如何使　Oracle的游标?
1.　oracle中的游标分为显　游标和隐式游标
2. 显　游标是　cursor...命令定义的游标，它可以对查询语句(select)返回的多条记录进　处理；隐式游标是在执　插　(insert)删除(delete)修改(update)和返回单条记录的查询(select)语句时由PL/SQL　动定义的。
3.显式游标的操作：打开游标、操作游标、关闭游标；PL/SQL 隐式地打开SQL 游标，并在它内部处理SQL 语句，然后关闭它

3. Oracle中function和procedure的区别?
1.可以理解函数是存储过程的　种
2.函数可以没有参数,但是　定需要　个返回值，存储过程可以没有参数,不需要返回值
3.函数return返回值没有返回参数模式，存储过程通过out参数返回值,如果需要返回多个参数则建议使　存储过程
4.在sql数据操纵语句中只能调　函数　不能调　存储过程

4. Oracle的导　导出有　种　式，有何区别?

1.使 oracle 具 exp/imp
2.使 plsql相关 具
  法1.导 /导出的是 进制的数据， 2.plsql导 /导出的是sql语句的 本 件

5. Oracle中有哪 种 件？
数据 件（ 般后缀为.dbf或者.ora）， 志 件(后缀名.log) 控制 件（后缀名为.ctl）

6.怎样优化Oracle数据库，有 种 式？
个 理解，数据库性能最关键的因素在于IO，因为操作内存是快速的，但是读写磁盘是速度很慢的，优化数据库最关键的问题在于减少磁盘IO，就个 理解应该分为物理的和逻辑的优化，物理的是指oracle产品本 的 些优化，逻辑优化是指应
物理优化的 些原则：
1. Oracle的运 环境（ 络，硬件等）
2.使 合适的优化器
3.合理配置oracle实例参数
4.建 合适的索引（减少IO）
5.将索引数据和表数据分开在不同的表空间上（降低IO冲突）
6.建 表分区，将数据分别存储在不同的分区上（以空间换取时间，减少IO）
  逻辑上优化：
1.可以对表进 逻辑分割，如中国移动 户表，可以根据 机尾数分成10个表，这样对性能会有 定的作
2. Sql语句使 占位符语句，并且开发时候必须按照规定编写sql语句（如全部 写，全部 写等）oracle解析语句后会放置到共享池中
如： select * from Emp where name=?这个语句只会在共享池中有 条， 如果是字符串的话，那就根据不同名字存在不同的语句，所以占位符效率较好
3.数据库不仅仅是 个存储数据的地 ，同样是 个编程的地 ， 些耗时的操作，可以通过存储过程等在 户较少的情况下执 ，从 错开系统使 的 峰时间，提 数据库性能
4.尽量不使 *号，如select * from Emp，因为要转化为具体的列名是要查数据字典， 较耗时
5.选择有效的表名
对于多表连接查询，可能oracle的优化器并不会优化到这个程度， oracle中多表查询是根据FROM 字句从右到左的数据进 的，那么最好右边的表（也就是基础表）选择数据较少的表，这样排序更快速，如果有 link表（多对多中间表），那么将link表
6. Where字句 规则
Oracle 中Where 字句时从右往左处理的，表之间的连接写在其他条件之前，能过滤掉 常多的数据的条件，放在 where 的末尾， 另外!=符号 较的列将不使 索引，列经过了计算（如变 写等）不会使 索引（需要建 起函数），is null，is not n

7.使 Exits Not Exits替代 In Not in
8.合理使 事务，合理设置事务隔离性
数据库的数据操作 较消耗数据库资源的，尽量使 批量处理，以降低事务操作次数
7. Oracle中字符串 什么符号链接？
Oracle中使 ||这个符号连接字符串 如 ‘abc’ || ‘d’
8. Oracle分区是怎样优化数据库的?
Oracle的分区可以分为：列表分区、范围分区、散列分区、复合分区。
1. 增强可 性：如果表的 个分区由于系统故障 不能使 ，表的其余好的分区仍可以使 ；
2. 减少关闭时间：如果系统故障只影响表的 部份分区，那么只有这部份分区需要修复，可能 整个 表修复花的时间更少；
3. 维护轻松：如果需要得建表，独产管理每个公区 管理单个 表要轻松得多；
4. 均衡I/O：可以把表的不同分区分配到不同的磁盘来平衡I/O改善性能；
5. 改善性能：对 表的查询、增加、修改等操作可以分解到表的不同分区来并 执 ，可使运 速度更快
6. 分区对 户透明，最终 户感觉不到分区的存在。
9. Oracle是怎样分页的?
Oracle中使 rownum 来进 分页,这个是效率最好的分页 法,hibernate也是使 rownum 来进 oralce分页的
```
select * from
  ( select rownum r,a from tabName where rownum <= 20 )
where r > 10
```
10. Oralce怎样存储 件，能够存储哪些 件？
Oracle 能存储 clob，nclob、 blob， bfile
Clob 可变长度的字符型数据，也就是其他数据库中提到的 本型数据类型
Nclob 可变字符类型的数据，不过其存储的是Unicode字符集的字符数据
Blob 可变长度的 进制数据
Bfile数据库外 存储的可变 进制数据
11. Oracle中使 了索引的列，对该列进 where 条件查询、分组、排序、使 聚集函数，哪些 到了索引？
均会使 索引， 值得注意的是复合索引（如在列 A 和列B 上建 的索引）可能会有不同情况
12.数据库怎样实现每隔30分钟备份 次？
通过操作系统的定时任务调 脚本导出数据库
13. Oracle中where 条件查询和排序的性能 较？
Order by使 索引的条件极为严格，只有满 如下情况才可以使 索引，
1.order by中的列必须包含相同的索引并且索引顺序和排序顺序 致
2.不能有null值的列
所以排序的性能往往并不 ，所以建议尽量避免order by
14.解释冷备份和热备份的不同点以及各 的优点？
冷备份发 在数据库已经正常关闭的情况下，将关键性 件拷贝到另外位置的 种说法
热备份是在数据库运 的情况下，采 归档 式备份数据的 法
冷备的优缺点：
1．是 常快速的备份 法（只需拷贝 件）
2．容易归档（简单拷贝即可）
3．容易恢复到某个时间点上（只需将 件再拷贝回去）
4．能与归档 法相结合，作数据库 "最新状态"的恢复。
5．低度维护， 度安全。
冷备份不 ：
1．单独使 时，只能提供到 "某 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 不能作其它 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上，速度会很慢。
4．不能按表或按 户恢复。

热备的优缺点

1．可在表空间或数据 件级备份，备份时间短。
2．备份时数据库仍可使 。
3．可达到秒级恢复（恢复到某 时间点上）。
4．可对 乎所有数据库实体作恢复。
5．恢复是快速的，在 多数情况下在数据库仍 作时恢复。
 热备份的不 是：
  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 于时间点的恢复。
  ．3因难于维护，所以要特别仔细 ，不允许"以失败 告终"。
15.解释data block ， extent和 segment的区别？
data block数据块，是oracle最 的逻辑单位，通常oracle从磁盘读写的就是块
extent区，是由若 个相邻的block组成
segment 段，是有 组区组成
tablespace表空间，数据库中数据逻辑存储的地 ， 个tablespace可以包含多个数据 件
16. 较truncate和delete命令 ？
1. Truncate和delete都可以将数据实体删掉，truncate的操作并不记录到 rollback 志，所以操作速度较快，但同时这个数据部能恢复
2. Delete操作不腾出表空间的空间
3. Truncate不能对视图等进 删除
4. Truncate是数据定义语 （DDL ）， delete是数据操纵语 (DML)
17.解释什么是死锁，如何解决Oracle中的死锁？
简 之就是存在加了锁 没有解锁，可能是使 锁没有提交或者回滚事务，如果是表级锁则不能操作表，客户端处于等在状态，如果是 级锁则不能操作锁定
解决办法：
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;
```
2.杀进程中的会话
```
alter system kill session "sid,serial#";
```
18.简述oracle中 dml、ddl、dcl的使
Dml 数据操纵语 ，如select，update、delete，insert
Ddl 数据定义语 ，如create table，drop table等等
Dcl 数据控制语 ， 如 commit、 rollback，grant、 invoke等
19.说说oracle中的经常使 到得函数
Length 长度、 lower 写、upper 写， to_date转化 期， to_char转化字符
Ltrim去左边空格、 rtrim去右边空格，substr取字串，add_month 增加或者减掉 份、to_number 转变为数字
20.怎样创建 个存储过程,游标在存储过程怎么使 ,有什么好处?
附：存储过程的 般格式，游标使 参考问题
1 .使 游标可以执 多个不相关的操作.如果希望当产 了结果集后,对结果集中的数据进 多种不相关的数据操作
2.使 游标可以提供脚本的可读性
3.使 游标可以建 命令字符串,使 游标可以传送表名,或者把变量传送到参数中,以便建 可以执 的命令字符串.
但是个 认为游标操作效率不太 ，并且使 时要特别 ，使 完后要及时关闭
存储过程优缺点：

优点：
1.存储过程增强了SQL 语　的功能和灵活性。存储过程可以　流控制语句编写，有很强的灵活性，可以完成复杂的判断和较复杂的运算。
2.可保证数据的安全性和完整性。
　3　通过存储过程可以使没有权限的　户在控制之下间接地存取数据库，从　保证数据的安全。
　　通过存储过程可以使相关的动作在　起发　，从　可以维护数据库的完整性。
3.再运　存储过程前，数据库已对其进　了语法和句法分析，并给出了优化执　　案。这种已经编译好的过程可极　地改善SQL 语句的性能。由于执　SQL 语句的　部分　作已经完成，所以存储过程能以极快的速度执　。
4.可以降低　络的通信量,不需要通过　络来传送很多sq语句到数据库服务器了
5.使体现企业规则的运算程序放　数据库服务器中，以便集中控制
　　当企业规则发　变化时在服务器中改变存储过程即可，　须修改任何应　程序。企业规则的特点是要经常变化，如果把体现企业规则的运算程序放　应　程序中，则当企业规则发　变化时，就需要修改应　程序　作量　常之　（修改、发
缺点：
1.可移植性差
2.占　服务器端多的资源，对服务器造成很　的压
3.可读性和可维护性不好

```
Create [or replace] procedure过程名字（参数 …) as
vs_ym_sn_end CHAR(6);     -同期终     份
CURSOR cur_1 IS   --定义游标(简单的说就是　个可以遍历的结果集)
SELECT area_code,CMCODE,SUM(rmb_amt)/10000 rmb_amt_sn,SUM(usd_amt)/10000 usd_amt_sn
FROM BGD_AREA_CM_M_BASE_T
  WHERE ym >= vs_ym_sn_beg
  AND ym <= vs_ym_sn_end
GROUP BY area_code,CMCODE;
BEGIN
-- 输　参数给变量赋初值，　到了 Oralce的SUBSTR TO_CHAR ADD_MONTHS TO_DATE          等很常　的函数。
vs_ym_beg := SUBSTR(is_ym,1,6);
vs_ym_end := SUBSTR(is_ym,7,6);
vs_ym_sn_beg := TO_CHAR(ADD_MONTHS(TO_DATE(vs_ym_beg,"yyyymm"), -12),"yyyymm");
vs_ym_sn_end := TO_CHAR(ADD_MONTHS(TO_DATE(vs_ym_end,"yyyymm"), -12),"yyyymm");
-先删除表中特定条件的数据。
DELETE FROM xxxxxxxxxxx_T WHERE ym = is_ym;
  -然后　内置的DBMS_OUTPUT　　对象的put_line 法打印出影响的记录　数，其中　到　个系统变量SQL%rowcount
DBMS_OUTPUT.put_line("del 上　记录="||SQL%rowcount||"条");

INSERT INTO xxxxxxxxxxx_T(area_code,ym,CMCODE,rmb_amt,usd_amt)
SELECT area_code,is_ym,CMCODE,SUM(rmb_amt)/10000,SUM(usd_amt)/10000
FROM BGD_AREA_CM_M_BASE_T
  WHERE ym >= vs_ym_beg
  AND ym <= vs_ym_end
GROUP BY area_code,CMCODE;

DBMS_OUTPUT.put_line("ins 当　记录="||SQL%rowcount||"条");
-遍历游标处理后更新到表。遍历游标有　种　法，　for语句是其中　较直观的　种。
FOR rec IN cur_1 LOOP
  UPDATE xxxxxxxxxxx_T
  SET rmb_amt_sn = rec.rmb_amt_sn,usd_amt_sn = rec.usd_amt_sn
   WHERE area_code = rec.area_code
   AND CMCODE = rec.CMCODE
   AND ym = is_ym;
END LOOP;
COMMIT;
-错误处理部分。OTHERS　表　除了声明外的任意错误。SQLERRM　是系统内置变量保存了当前错误的详细信息。
EXCEPTION
   WHEN OTHERS THEN
      vs_msg := "ERROR IN xxxxxxxxxxx_p("||is_ym||"):"||SUBSTR(SQLERRM,1,500);
   ROLLBACK;
   把当前错误记录进　志表。
   INSERT INTO LOG_INFO(proc_name,error_info,op_date)
   VALUES("xxxxxxxxxxx_p",vs_msg,SYSDATE);
   COMMIT;
   RETURN;
END;
```

21.怎样创建　个　个索引,索引使　的原则,有什么优点和缺点
创建标准索引：
CREATE　INDEX　索引名 ON 表名 列名）　TABLESPACE　表空间名;
创建唯　索引:
CREATE unique INDEX　索引名 ON 表名 列名）　TABLESPACE　表空间名;
创建组合索引:
CREATE INDEX　索引名 ON 表名 列名1,列名2）　TABLESPACE　表空间名;
创建反向键索引:
CREATE INDEX　索引名 ON 表名 列名）reverse TABLESPACE　表空间名;
索引使　原则：
索引字段建议建　NOT NULL　约束
经常与其他表进　连接的表，在连接字段上应该建　索引；
经常出现在Where　句中的字段且过滤性很强的，特别是　表的字段，应该建　索引；
可选择性　的关键字 ，应该建　索引；
可选择性低的关键字，但数据的值分布差异很　时，选择性数据　较少时仍然可以利　索引提　效率
复合索引的建　需要进　仔细分析；尽量考虑　单字段索引代替：
A、正确选择复合索引中的第　个字段，　般是选择性较好的且在 where　句中常　的字段上；
B、复合索引的　个字段经常同时以 AND　式出现在Where　句中可以建　复合索引；否则单字段索引；
C、如果复合索引中包含的字段经常单独出现在 Where　句中，则分解为多个单字段索引；
D、如果复合索引所包含的字段超过 3个，那么仔细考虑其必要性，考虑减少复合的字段；
E、如果既有单字段索引，　有这　个字段上的复合索引，　般可以删除复合索引；
频繁DML 的表，不要建　太多的索引；
不要将那些频繁修改的列作为索引列；
索引的优缺点：
有点：
1.创建唯　性索引，保证数据库表中每　　数据的唯　性
2.　　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　　特别有意义。
4.在使　分组和排序　句进　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　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　个索引还要占　定的物理空间，如果要建　聚簇索引，那么需要的空间就会更
4.当对表中的数据进　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

22.怎样创建　个视图,视图的好处,视图可以控制权限吗?
create view视图名 as selec列名 别名]　…　fro表 [unio [all] select …]]
好处：
1.可以简单的将视图理解为sql查询语句，视图最　的好处是不占系统空间
2.　些安全性很　的系统，不会公布系统的表结构，可能会使　视图将　些敏感信息过虑或者重命名后公布结构
3.简化查询
可以控制权限的，在使　的时候需要将视图的使　权限grant给　户
23.怎样创建　个触发器,触发器的定义,触发器的游标怎样定义
CREATE [OR REPLACE] TIGGER　　触发器名 触发时间 触发事件
  ON 表名
  [FOR EACH ROW]
  BEGIN
    pl/sq语句
    CURSOR　游标名 is　SELECT * FROM 表名 （定义游标）
    END
  其中：
  触发器名：触发器对象的名称。
  由于触发器是数据库　动执　的，因此该名称只是　个名称，没有实质的　途。
触发时间：指明触发器何时执　，该值可取：
before—表　在数据库动作之前触发器执　；
after—表　在数据库动作之后出发器执　。
触发事件：指明哪些数据库动作会触发此触发器：
　　insert 数据库插　会触发此触发器;

24. oracl创建表的　种　式,应该注意些什么
不知道这个题　是不是记错了，感觉很怪
1.使　图形　具创建表

2.使 数据ddl语句创建表
3.可以在plsql代码中动态创建表
应该注意：是否有创建表的权限，使 什么表空间等
25.怎样将 个旧数据库数据移到 个新的数据库
1. Imp/exp将数据库中的数据导 到新的库中
2.如果是存储迁移直接将存储设备挂到新机器上
26.主键有 种;
字符型，整数型、复合型
27. oracl的锁 种,定义分别是什么;
1. 共享锁 (ROW SHARE)
2. 排他锁(ROW EXCLUSIVE)
3 .共享锁(SHARE)
4. 共享 排他锁(SHARE ROW EXCLUSIVE)
5. 排他锁(EXCLUSIVE)
使 法：
SELECT * FROM order_master WHERE vencode="V002"
FOR UPDATE WAIT 5;
LOCK TABLE order_master IN SHARE MODE;
LOCK TABLE itemfile IN EXCLUSIVE MODE NOWAIT;
ORACLE 锁具体分为以下 类：
1.按 户与系统划分，可以分为 动锁与显 锁
  动锁：当进 项数据库操作时，缺省情况下，系统 动为此数据库操作获得所有有必要的锁。
显 锁：某些情况下，需要 户显 的锁定数据库操作要 到的数据，才能使数据库操作执 得更好，显 锁是 户为数据库对象设定的。
2 .按锁级别划分，可分为共享锁与排它锁
共享锁：共享锁使 个事务对特定数据库资源进 共享访问——另 事务也可对此资源进 访问或获得相同共享锁。共享锁为事务提供 并发性，但如拙劣的事务设计+共享锁容易造成死锁或数据更新丢失。
排它锁：事务设置排它锁后，该事务单独获得此资源，另 事务不能在此事务提交之前获得相同对象的共享锁或排它锁。
3.按操作划分，可分为DML 锁、DDL 锁
DML 锁 可以分为， 锁、表锁、死锁
  锁：当事务执 数据库插 、更新、删除操作时，该事务 动获得操作表中操作 的排它锁。
表级锁：当事务获得 锁后，此事务也将 动获得该 的表锁(共享锁),以防 其它事务进 DDL 语句影响记录 的更新。事务也可以在进 过程中获得共享锁或排它锁，只有当事务显 使 LOCK TABLE 语句显 的定义 个排它锁时，事务才会获得
死锁：当两个事务需要 组有冲突的锁， 不能将事务继续下去的话，就出现死锁。
如事务1在表A 记录#3中有 排它锁，并等待事务2在表A中记录#4中排它锁的释放， 事务2在表A记录 #4中有 排它锁，并等待事务；1在表A中记录#3中排它锁的释放，事务1与事务2彼此等待，因此就造成了死锁。死锁 般是因拙劣的事务设计
DDL 锁 可以分为：排它DDL 锁、共享DDL 锁、分析锁
排它DDL 锁：创建、修改、删除 个数据库对象的 DDL 语句获得操作对象的 排它锁。如使 alter table语句时，为了维护数据的完成性、 致性、合法性，该事务获得 排它DDL 锁。
共享DDL 锁：需在数据库对象之间建 相互依赖关系的 DDL 语句通常需共享获得DDL 锁。
如创建 个包，该包中的过程与函数引 了不同的数据库表，当编译此包时，该事务就获得了引 表的共享 DDL 锁。
分析锁：ORACLE 使 共享池存储分析与优化过的SQL 语句及PL/SQL 程序，使运 相同语句的应 速度更快。 个在共享池中缓存的对象获得它所引 数据库对象的分析锁。分析锁是 种独特的DDL 锁类型，ORACLE 使 它追踪共享池对象及它所
4.内部闩锁
内部闩锁：这是ORACLE 中的 种特殊锁， 于顺序访问内部系统结构。当事务需向缓冲区写 信息时，为了使 此块内存区域，ORACLE 先必须取得这块内存区域的闩锁，才能向此块内存写 信息。

29. rowid,rownum的定义
1. rowid和rownum 都是虚列
2. rowid是物理地址， 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，从下 的例 可以看出其中的区别。
30. oracle中存储过程，游标和函数的区别
游标类似指针，游标可以执 多个不相关的操作.如果希望当产 了结果集后,对结果集中的数据进 多 种不相关的数据操作
函数可以理解函数是存储过程的 种；函数可以没有参数,但是 定需要 个返回值，存储过程可以没有参数,不需要返回值；两者都可以通过out参数返回值,如果需要返回多个参数则建议使 存储过程；在sql数据操纵语句中只能调 函数 不能调
31.使 oracle伪列删除表中重复记录：
Delete table t where t.rowid!=(select max(t1.rowid) from table1 t1 where t1.name=t.name)

使 scott/tiger户下的emp 表和dept表完成下列练习，表的结构说明如下
emp 员 表(empno 员 号/ename 员 姓名/job 作/mgr上级编号/hiredat受雇 期/sa薪 /comm 佣 /deptno部门编号)
dept部门表(deptno部门编号/dname 部门名称/lo地点)
  资 = 薪 + 佣
1．列出 少有 个员 的所有部门。
2．列出薪 “SMITH”多的所有员 。
3．列出所有员 的姓名及其直接上级的姓名。
4．列出受雇 期早于其直接上级的所有员 。
5．列出部门名称和这些部门的员 信息，同时列出那些没有员 的部门。
6．列出所有“CLERK”（办事员）的姓名及其部门名称。
7．列出最低薪 于1500 的各种 作。
8．列出在部门“SALES”（销售部） 作的员 的姓名，假定不知道销售部的部门编号。
9．列出薪 于公司平均薪 的所有员 。
10．列出与“SCOTT”从事相同 作的所有员 。
11．列出薪 等于部门30中员 的薪 的所有员 的姓名和薪 。
12．列出薪 于在部门30 作的所有员 的薪 的员 姓名和薪 。
13．列出在每个部门 作的员 数量、平均 资和平均服务期限。
14．列出所有员 的姓名、部门名称和 资。
15．列出所有部门的详细信息和部门 数。
16．列出各种 作的最低 资。
17．列出各个部门的MANAGER （经理）的最低薪 。
18．列出所有员 的年 资,按年薪从低到 排序。解答：
1．列出 少有 个员 的所有部门。(两个表联合查询，及group by…having的 法)
select dname from dept where deptno in(select deptno from emp group by deptno having count(*)>1);
2．列出薪 “SMITH”多的所有员 。(经典的 连接查询)
select ename from emp where sal>(select sal from emp where ename like’SMITH’);
3．列出所有员 的姓名及其直接上级的姓名。(多次对 查询,为表的取个别名，内部查询可以像对象 样引 外部的对象的字段，这 引 与编程中的作 域相似，即与{类
)
select ename,(select ename from emp where empno in(a.mgr)) from emp a ;
4．列出受雇 期早于其直接上级的所有员 。(同上，期可直接拿来 较)
select ename from emp a where HIREDATE<(select HIREDATE from emp where empno in(a.mgr));
5．列出部门名称和这些部门的员 信息，同时列出那些没有员 的部门(以emp 表为主，左连接查询)
select dname,emp.* from dept left join emp on dept.deptno=emp.deptno;
6．列出所有“CLERK”（办事员）的姓名及其部门名称。(域，注意())
select ename,(select dname from dept where deptno in(a.deptno)) as dname from emp a where JOB like’CLERK’;
7．列出最低薪 于1500 的各种 作。
select job from emp where sal>1500;
8．列出在部门“SALES”（销售部） 作的员 的姓名，假定不知道销售部的部门编号。(经典的两个表连接)
select ename from emp where deptno=(select deptno from dept where dname like’SALES’);
9．列出薪 于公司平均薪 的所有员 。(反复查)
select ename from emp where sal>(select avg(sal) from emp);
10．列出与“SCOTT”从事相同 作的所有员 。(排除)
select ename from emp where job in(select job from emp where ename like’SCOTT’) and ename!=’SCOTT’ ;
11．列出薪 等于部门30中员 的薪 的所有员 的姓名和薪 。(any的 法，且排挤)
select ename,sal from emp where sal=any(select sal from emp wheredeptno=30) and deptno!=30;
12．列出薪 于在部门30 作的所有员 的薪 的员 姓名和薪 。(max的 法)
select sal,ename from emp where sal>(select max(sal) from emp where deptno=30);

13．列出在每个(每个是关键字,对此group by)部门 作的员 数量、平均 资和平均服务期限。(经典的group by 法)select deptno,count(*),avg(a.sal),avg(sysdate-HIREDATE)
from emp a group by deptno;
14．列出所有员 的姓名、部门名称和 资.(经典的两个表的连接查询， 具体的名称替换 个表中的主键的 id 解决很多 在实际运 中会遇到的不能绑定多列的问题)，也可
where 来查询 ,与题5 较)
select ename,sal,(select dname from dept a where a.deptno=b.deptno)as dname from emp b;
15．列出所有部门的详细信息和部门 数。(因为是*，将显 dept和后 临时表b的全部字段(注意:不只是dept的字段,注意*号))
select * from dept a left join (select deptno,count(*) from emp group by deptno) b on a.deptno=b.deptno ;

16．列出各种(与每个同义(参看题13)) 作的最低 资。
select job,min(sal) from emp group by job ;

17．列出各个部门的MANAGER （经理,经理唯 ，不 group by）的最低薪 。select min(sal) from emp where job like’MANAGER’因为MANAGER 是值不是字段，所以不能写)

18．列出所有员 的年 资,按年薪从低到 排序。(nvl空转化函数)
select ename,(sal+nvl(comm,0))*12 as sal from emp order by sal ;

户创建、修改、删除

户创建

create user username profile default identified by “user_passwd” default tablespace

tablespace_name account unlock;

修改 户密码：

alter user username identified by char_name

修改 户使 状态

alter user username account unlock

户赋权限：

grant connect,resource,select any table,update any table,delete any table,insert any

table,select

any dictionary,create any procedure,execute any procedure,create any TRIGGER,create

any view, unlimited tablespace

drop any view,create any sequence,select any sequence,drop any sequence,CREATE

DATABASE LINK,

CREATE PUBLIC DATABASE LINK,DROP PUBLIC DATABASE LINK,CREATE

ANY synonym,DROP ANY synonym,

CREATE PUBLIC synonym,DROP PUBLIC SYNONYM,SELECT_CATALOG_ROLE

to &u;

给某 个 户赋予某张表的某种权限：

grant privilege_name(select insert update)table_name to username;

例如：

grant update ON TABLE_NAME TO username;

grant delete ON TABLE_NAME TO username;

grant insert ON TABLE_NAME TO username;

户权限回收：

revoke role_name from username;

alter user username default role all;

户删除：

Drop user username cascade （如果有数据，带 cascade 参数）

查看 户属性和状态：

select * from dba_users;

select * from dba_sys_privs

select * from dba_tab_privs

权限名称：

administer database trigger

administer resource manager

alter any cluster

alter any dimension

alter any evaluation context

alter any index

alter any indextype

alter any library

alter any outline

alter any procedure

alter any role

alter any rule

alter any rule set

alter any sequence

alter any snapshot

alter any table

alter any trigger

alter any type

alter database

alter profile

alter resource cost

alter rollback segment

alter session

alter system

alter tablespace

alter user

analyze any

audit any

audit system

backup any table

debug any procedure

debug connect session

delete any table

dequeue any queue

drop any cluster

drop any context

drop any dimension

drop any directory

drop any evaluation context

drop any index

drop any indextype

drop any library

drop any operator

drop any outline

drop any procedure

drop any role

drop any rule

drop any rule set

drop any sequence

drop any snapshot

drop any synonym

drop any table

drop any trigger

drop any type

drop any view

drop profile

drop public database link

drop public synonym

drop rollback segment

drop tablespace

become user

comment any table

create any cluster

create any context

create any dimension

create any directory

create any evaluation context

create any index

create any indextype

create any library

create any operator

create any outline

create any procedure

create any rule

create any rule set

create any sequence

create any snapshot

create any synonym

create any table

create any trigger

create any type

create any view

create cluster

create database link

create dimension

create evaluation context

create indextype

create library

create operator

create procedure

create profile

create public database link

create public synonym

create role

create rollback segment

create rule

create rule set

create sequence

create session

create user

create view

drop user

enqueue any queue

execute any evaluation context

execute any indextype

execute any library

execute any operator

execute any procedure

execute any rule

execute any rule set

execute any type

exempt access policy

flashback any table

force any transaction

force transaction

global query rewrite

grant any object privilege

grant any privilege

grant any role

insert any table

lock any table

manage any queue

manage tablespace

on commit refresh

query rewrite

restricted session

resumable

select any dictionary

select any sequence

select any table

under any table

under any type

under any view

unlimited tablespace

update any table

create snapshot

create synonym

create table

create tablespace

create trigger

create type

2、　　创建、修改、删除

　　创建

create role role_name;

　　权限修改

grant privilege_name to role_name

revoke privilege_name from role_name

　　删除

drop role role_name

系统　　：

AQ_ADMINISTRATOR_ROLE

AQ_USER_ROLE

AUTHENTICATEDUSER

CONNECT

CTXAPP

DBA

DELETE_CATALOG_ROLE

EJBCLIENT

EXECUTE_CATALOG_ROLE

EXP_FULL_DATABASE

GATHER_SYSTEM_STATISTICS

HS_ADMIN_ROLE

WKUSER

WM_ADMIN_ROLE

XDBADMIN

3、权限增加和收回

IMP_FULL_DATABASE

JAVADEBUGPRIV

JAVAIDPRIV

JAVASYSPRIV

JAVAUSERPRIV

JAVA_ADMIN

JAVA_DEPLOY

LOGSTDBY_ADMINISTRATOR

OEM_MONITOR

RECOVERY_CATALOG_OWNER

RESOURCE

SALES_HISTORY_ROLE

SELECT_CATALOG_ROLE

grant privilege_name to role_name

revoke privilege_name from role_name

1、单表创建

直接创建

Create table table_name

(

（增加权限）

（收回权限）

字段 1类型 1,

字段 2类型 2,

、、、、、

字段 n类型 n

) tablespace_name;

创建类似表

Create table table_name as select * from table_name1;

创建表结构：

Create table table_name as select * from table_name where 字段 1=字段 2;

表查询：

Select * from table_name;

Select * from table_name where条件

Select字段 1别名 1，字段 2

别名 2，字段 n别名 n from table_name

字段中可以进　代数运算，包括：＋－×÷countsunavgetrunkround等函数

表插　：

Inser字段 1，字段 2、、字段 n into table_name valu字段值 1，字段值 2、、字段值 n ；

循环插　：

begin

for i in 1..10 loop

insert into table_name values (…);

end loop;

end;

表更新：

Update table_name set字段 1 where条件

表删除：

drop table table_name

表记录删除：

delete table_name where条件；

truncate table table_name清空表记录，保留表结构

表名修改：

rename table_name1 to table_name2

注意：表名修改后，索引还在

表迁移，（从　个表空间迁移到另外　个表空间）

alter table table_name move tablespace tablespace_name nologging parallele 4 ;

注意：将表从　个表空间迁移到另外　个表空间，必须进　索引重建，存储过程、触发器、其他程序包都需要编译，以免执　报错。例如：

表压缩：

不同表空间： alter table table_name move tablespace tablespace_name compress;

相同表空间：alter table table_name move compress;

表统计信息收集

exec dbms_stats.gather_table_stats(ownname => ‘username’,tabname =>’ table_name’,degree =>10,cascade =>true,estimate_percent =>25);

例如：

exec dbms_stats.gather_table_stats(ownname => ‘test’,tabname

=>’data_table1_name1’,degree =>4,cascade =>true,estimate_percent =>30);

select a.row2_id, sum(nvl(b.item_value, 0))

from zk.name _201001 a, zk.table_name _201001 b

where a.so_row1 = b.so_row1 and a.row_a in (1, 4, 5) and a.row2_id = XXXXXXX and

book_row3_id in (NNNNNXXX1, NNNNNXXX2, NNNNNXXX3, NNNNNXXX4,

NNNNNXXX5, NNNNNXXX6) group by a.row _id;

表语句执 很慢，检查执 计划，zk.table_name_201001 b不引 索引，做表分析、重建索引都

效，执 计划中均没有引 索引，最后强制使 索引效果明显，执 计划改变，索引引 。效果明

显

select /*+ index(b pk_table_name_201001)*/

a.serv_id, sum(nvl(b.item_value, 0))

from zk.name_201001 a, zk. table_name _201001 b

where a. so_row1 = b.so_row1 and a.busi_code in (1, 4, 5) and a.row2_id = XXXXXXX and

book_row3_id in (NNNNNXXX1, NNNNNXXX2, NNNNNXXX3, NNNNNXXX4,

NNNNNXXX5, NNNNNXXX6)

group by a.row2_id;

2、分区表创建

为了使 量的数据在读写操作和查询中速度更快，Oracle提供了对表和索引进 分区的技术，

以改善 型应 系统的性能。使 分区的优点：

A、增强可 性：如果表的某个分区出现故障，表在其他分区的数据仍然可 ；

B、维护 便：如果表的某个分区出现故障，需要修复数据，只修复该分区即可；

C、均衡I/O：可以把不同的分区映射到磁盘以平衡 I/O，改善整个系统性能；

D、改善查询性能：对分区对象的查询可以仅搜索 关 的分区，提 检索速度

Oracle数据库提供对表或索引的分区 法有三种：

1、范围分区

2、Hash 分区（散列分区）

3、列表分区

4、范围—散列分区

5、范围—列表分区

2.1、范围分区表

分区表表创建： 般都是按照时间来创建

```
create table table_name
(
row1 char1 not null,
row2 char2 not null,
、、、、、、
rown  number not null
)
partition by range (paration_row)
(
partition partition_name1 values less than (to_date( ‘value_name’ ,  ‘yyyy-mm-dd’ ,
 ‘nls_calendar=gregorian’ )) tablespace tablespace_name1,
partition partition_name2 values less than (to_date( ‘value_name’ ,  ‘yyyy-mm-dd’ ,
 ‘nls_calendar=gregorian’ )) tablespace tablespace_name2,
、、、、、、、、、、
partition partition_namen values less than (to_date( ‘value_name’ ,  ‘yyyy-mm-dd’ ,
 ‘nls_calendar=gregorian’ )) tablespace tablespace_namen,
partition partmax values less than (maxvalue)
tablespace ora_data
pctfree 10
initrans 16
maxtrans 255
storage
(
initial 64k
minextents 1
maxextents unlimited ))
```

表插 ：

```
Insert into table_name1 as select * from table_name2
```

表分区删除：

alter table 表名 truncate partition 分区名称 drop storage;

注意：分区删除以后，需要重新创建索引。

分区分割：

```
alter table table_name split partition partmax at(to_date( ‘values_name’ ,  ‘yyyy-mm-dd’ )) into
(partition prtition_namexx tablespace tablespace_name , partition partmax);
```

分区查询：

```
select * from table_name partition (partition_name);
```

2.2、hash 分区表（散列分区）

Hash 分区表创建：hash 分区表的分区数量 般是 2的 n次幂，这样记录分布在各个分区上就

 较均匀，可以进 I/O的均衡。

```
create table table_name
(
```

```
row1 char1 not null,
row2 char2 not null,
、、、、、、
rown  NUMBER not null
)
partition by hash (paration_row)
(
partition partition_name1 tablespace tablespace_name1,
partition partition_name2 tablespace tablespace_name2,
partition partition_name3 tablespace tablespace_name3
、、、、、、、、、、
partition partition_nameX tablespace tablespace_nameX
)
```

表插 ：

```
insert into table_name1 as select * from table_name2
```

分区查询：

Select * From table_name Partition (partition_name)分区表 般由于数据分布均衡性，查询不是

通过对分区的操作进 的。update、inser操作同普通表。

2.3、列表分区表

散列分区表创建：

```
create table table_name
(
row1 char1 not null,
row2 char2 not null,
、、、、、、
rown  number not null
)
partition by list (paration_row)
(

partition   partition_name1

values

(row_value1,row_value3,row_value8)

tablespace

tablespace_name1,

partition   partition_name2

values

(row_value4,row_value6,row_value7)

tablespace

tablespace_name2,
```

partition

partition_name3

values

(row_value9,row_value10,row_value11,row_value12,row_value15)

tablespace tablespace_name3

、、、、、、、、、、

partition partition_nameX values (default) tablespace tablespace_nameX

)

表的插 、更新、删除和普通表相同，在 hash分区和 list分区中， 较困难的操作是：从 张 5千万以上的表中，要删除 千万条记录 较困，可以通过以下 法进 清理：

第 种 法：

a创建中间表和表备份， b进 数据插 ， c进 锁表， d进 表名修改， e进 索引创建和存储过程编译， f进 表 truncate操作

第 种 法：直接创建 job进 delete删除：这样长期做，会降低表的执 效果

a进 备份数据，b进 表记录删除，每次删除 1000——2000 条记录，不然会发 锁表，c重建索引，d进 表分析。

declare

v_lognum number; 数据库中拥有的 志 件数

v_needarc number; 需要归档的 志 件数

begin

select count(1) into v_lognum from v$log;

loop

loop

select count(1) into v_needarc from v$archive;

if v_needarc < v_lognum - 1 then

exit;

end if;

end loop;

delete from

对象.table_name where条件 1 and条件 2 and rownum<1000;

if sql%rowcount = 0 then

exit;

end if;

commit;

end loop;

end;

- - - - - -————————— —— -

create or replace procedure procedure_name(exp_date1 varchar2 ,exp_date2 varchar2)

as

del_sql varchar2(1024);

v_c integer;

begin

del_sql:=’ delete 对象.表名 a where exists (select 1 对象.表名 b where

条件 1 and

条件 2 and、、、

条件 n) and rownum<1000 ';

for i in 1..10000 loop

execute immediate del_sql using exp_date1,exp_date2;

if sql%rowcount = 0 then

exit;

end if;

commit;

end loop;

end procedure_name;

3、分区操作

添加分区

alter table table_partition_name add partition partition_name values less than

(to_date(‘row_values’,’yyyy-mm-dd’));

注意：以上添加的分区界限应该 于最后 个分区界限。

alter table table_partition_name

modify partition partition_name

add

subpartition

subpartition_name

values(‘row_values’);

删除分区

以下代码删除表分区：

alter table table_partition_name drop partition partition_name storage;

注意：如果删除的分区是表中唯 的分区，那么此分区将不能被删除，要想删除此分区，必须删除表

截断分区

截断某个分区是指删除某个分区中的数据，并不会删除分区，也不会删除其它分区中的数据。当表中即使只有 个分区时，也可以截断该分区。通过以下代码截断分区：

alter table table_partition_name truncate partition partition_name;

合并分区

合并分区是将相邻的分区合并成 个分区，结果分区将采 较 分区的界限，值得注意的是，不能将分区合并到界限较低的分区。以下代码实现了 partition_name1与 partition_name2分区的合并：

alter table table_partition_name merge partitions partition_name1,partition_name2 into partition partition_name2;

拆分分区

拆分分区将 个分区拆分两个新分区，拆分后原来分区不再存在。注意不能对 HASH 类型的分区进 拆分。

alter table table_partition_name sblit partition partition_name1

at(to_date(‘row_value’ ,’yyyy-mm-dd’))

into (partition partition_name1,partition partition_name2);

接合分区(coalesca)

结合分区是将散列分区中的数据接合到其它分区中，当散列分区中的数据 较 时，可以增加散列分区，然后进 接合，值得注意的是，接合分区只能 于散列分区中。通过以下代码进 接合分区：

alter table table_hash_partition_name coalesca partition;

重命名表分区

以下代码将 partition_name更改为 partition_name2

alter table table_partition_name rename partition partition_name1 to partition_name2;

相关查询

跨分区查询

select sum(*) from

(select count(*) cn from table_partition_name partition (partition_name1)

union all

select count(*) cn from table_partition_name partition (partition_name2));

查询表上有多少分区

select * from user_tab_partitions where table_name =’ table_partition_name’

查询索引信息

select object_name,object_type,tablespace_name,sum(value)

from v$segment_statistics

where statistic_name in (‘physical reads’ ,’physical write’ ,’logical reads’)and object_type=’ index’

group by object_name,object_type,tablespace_name order by 4 desc

-显　数据库所有分区表的信息：

select * from dba_part_tables

-显　当前　户可访问的所有分区表信息:

select * from all_part_tables

-显　当前　户所有分区表的信息：

select * from USER_PART_TABLES

-显　表分区信息显　数据库所有分区表的详细分区信息：

select * from dba_tab_partitions

-显　当前　户可访问的所有分区表的详细分区信息：

select * from all_tab_partitions

-显　当前　户所有分区表的详细分区信息：

select * from user_tab_partitions

-显　　分区信息显　数据库所有组合分区表的　分区信息：

select * from dba_tab_subpartitions

-显　当前　户可访问的所有组合分区表的　分区信息：

select * from all_tab_subpartitions

-显　当前　户所有组合分区表的　分区信息：

select * from user_tab_subpartitions

-显　分区列显　数据库所有分区表的分区列信息：

select * from dba_part_key_columns

-显　当前　户可访问的所有分区表的分区列信息：

select * from all_part_key_columns

-显　当前　户所有分区表的分区列信息：

select * from user_part_key_columns

-显　　分区列显　数据库所有分区表的　分区列信息：

select * from dba_subpart_key_columns

-显　当前　户可访问的所有分区表的　分区列信息：

select * from all_subpart_key_columns

-显　当前　户所有分区表的　分区列信息：

select * from user_subpart_key_columns

-怎样查询出 oracl数据库中所有的的分区表

select * from user_tables a where a.partitioned=’ yes’

-删除 个表的数据是

truncate table table_name;

-删除分区表 个分区的数据是

alter table table_partition_name truncate partition partition_nameN drop storage;

4、表状态查询

Select * From Dba_Tab_Partitions Where Table_Name=’装名%’;

Select owner,index_name,status,degree,table_name from dba_indexes where

table_name=’ table_name ’ ;

select owner,bytes/1024/1024,segment_nam,segment_type,tablespace_name from dba_segments

where

segment_name=’ table_name’ and segment_type=’ TABLE’ ;

5、DDL 语句操作

创建表

基本语法：

create [global temporary] table table_name(

column_name type [constraint constraint_def default default_exp]

[,column_name type [constraint constraint_def default default_exp] …])

[on commit {delete | preserve} rows]

tablespace tab_space;

其中：

1)global temporar说明改表的 都是临时的，这种表就称为临时表。

 的有效期由 on commit字句指定。临时表对于所有的会话都是可见的，但是这些 则是特定于

某个会话的。

2)table_name指定了要分配给该表的名称。

3)column_name 指定了要分配给某个列的名称。

4)type指定了对某个列的类型。

5)constraint_d指定了对某个列的约束的定义。

6)default_de指定了 个表达式， 来为某个列赋予默认值。

7)on commit控制临时表中 的有效期。delete说明这些 在事务的末尾要被删除。

preserve说明这些 在会话的末尾要被删除。若对临时表没有指定 on commit选项，那末默认值是

delete。

8)tab_space为该表指定表空间。若没有指定表空间，该表就被存储在该 户的默认表空间中。

获得有关表的信息可以通过如下操作获取有关表的信息：

对表执 describe命令。

desc order_status_temp

-注意 desc是 SQL*PLUS 命令，在 SQL 中不能执 。

查询 user_tables 它是数据字典的 部分。另：查询 户可访问的表的相关信息，可以查询

all_tables

```
select table_name, tablespace_name, temporary
from user_tables
where table_name in (‘table_name’, ‘ORDER_STATUS_TEMP’);
```

获得表中列的信息

从 user_tab_columns中可以获得有关表中各列的信息,另：通过访问 all_tab_columns可以获得有关可以访问的表中所有列的信息。

```
select column_name, data_type, data_length, data_precision, data_scale
from user_tab_columns
where table_name = ‘table_name’;
```

修改表

alter table语句可以 于对表进 修改。

alter table语句可以执 以下任务：

1)添加、修改、删除列；

2)添加或删除约束；

3)启 或禁 约束。

添加列

```
alter table table_name add modified_by integer;
alter table table_name add rowname类型 [date default sysdate not null];
```

修改列

1)修改列的长度，条件是该列的类型的长度可以修改， 如 char或 varchar2

2)修改数字列的精度；

3)修改列的数据类型；

4)修改列的默认值。

修改列的长度

```
alter table table_name modify rowname类型(长度);
```

只有在表中还没有任何 或所有列都为空值的情况下才可以减 列的长度。

修改数字列的精度

```
alter table table_name modify colname类型;
```

只有在表中还没有任何 或列为空值时才可以减 数字列的精度。

修改列的数据类型

```
alter table table_name modify colname类型;
```

若 个表中还没有任何 或列为空值，就可以将列修改为任何 种数据类型。否则，就只能将列的数据类型修改为 种兼容的数据类型。

例如，可以将 varchar2类型修改为 char，条件是没有缩短列的长度；但是不能将 date修改为 number 。

修改列的默认值

alter table table_name modify column column_name;

默认值只适　于新插　表中的　。

删除列

alter table table_name drop column column_name;

重命名表

rename 语句可以　于对表进　重命名。

rename table_name1 table_name2;

删除表数据，保留表结构

truncate table table_name

视图是基于　张表或多张表或另外　个视图的逻辑表。视图不同于表，视图本　不包含任何数据。表是实际独　存在的实体，是　于存储数据的基本结构。　视图只是　种定义，对应　个查询语句。视图的数据都来　于某些表，这些表被称为基表。通过视图来查看表，就像是从不同的　度来观察　个(或多个)表。

视图有如下　些优点：

a可以提　数据访问的安全性，通过视图往往只可以访问数据库中表的特定部分，限制了　户访问表的全部　和列。

b简化了对数据的查询，隐藏了查询的复杂性。视图的数据来　　个复杂的查询，　户对视图的检索却很简单。

c　个视图可以检索多张表的数据，因此　户通过访问　个视图，可完成对多个表的访问。

d视图是相同数据的不同表　，通过为不同的　户创建同　个表的不同视图，使　户可分别访问同　个表的不同部分。视图可以在表能够使　的任何地　使　，但在对视图的操作上同表相　有些限制，特别是插　和修改操作。对视图的操作将传递到基表，所以在表上定义的约束条件和触发器在视图上将同样起作　。

1、视图创建

create [or replace] [force|noforce] view view_name

[(alias[,alias]…)]

as subquery

[with check option [constraint constraint]]

[with read only]

例如：

创建视图 empv，该属兔仅包含 10部门雇员的细节信息。

create view empv

as select empno,ename,job

from emp

where deptno=10;

别名的使

例 1: create or replace view salv

as select empno employee_id,ename name,sal salary

from emp where deptno=30;

例 2: create or replace view salv

(employee_id,name,salary)

as select empno,ename,sal

from emp

where deptno=30;

创建复杂视图

例：创建　个基于两个表并且含有组函数的复杂视图

create or replace view dept_sum_v(name,minsal,maxsal,avgsal)

as select d.dname,min(e.sal),max(e.sal),avg(e.sal)

from emp e,dept d where e.deptno=d.deptno group by d.dname;

2、视图操作

查看视图

数据字典 USER_VIEWS

select view_name,text from user_views;

从视图检索数据

select * from salv;

视图上执　DML 操作

a)如果视图包含以下内容，则不能实现对数据的修改:

-GROUP　函数、GROUP BY　　句、DISTINCT 关键字

-使　表达式定义的列

-ROWNUM　伪列

b)如果视图中包含以下内容则不能删除数据

-GROUP　函数

-GROUP BY　　句

-DISTINCT 关键字

-ROWNUM　伪列

WITH CHECK OPTION　　句

-如果要确保在视图上执　的插　、更新操作仅限于　定的范围，便可使　WITH CHECK

OPTION　句

例:

create or replace view empv

as select * from emp where deptno=20

with check option constraint empv_ck;

测试 ：update empv set deptno=10 where empno=7369

结果：

ERROR 位于第 1 ：

ORA-01402: 视图 WITH CHECK OPTIDN 违反 where 句

测试 ：update empv set sal=2000 where empno=7369

结果：已更新 1 。

拒绝 DML 操作

-在视图定义时使 WITH READ ONLY 选项可以确保不能对视图执 DML 操作

例：

create or replace view empv(employee_id,employ_name,job_title)

as select empno,ename,job from emp where deptno=10

with read only;

3、视图删除

删除视图并不会删除数据，因为视图是基于数据库中的基表

DROP VIEW view_name;

例：DROP VIEW empv;

Oracle的"索引"对象，与表关联的可选对象，提 SQL 查询语句的速度;

索引直接指向包含所查询值的 的位置，减少磁盘I/O;

与所索引的表是相互独 的物理结构 ;

Oracle 动使 并维护索引，插 、删除、更新表后， 动更新索引 ;

语法：CREATE INDEX index ON table (column[, column]…);

B-tree结构(bitmap)

1、单表索引

索引创建：

create [unique | bitmap] index index_name on table_name(col_name) pctfree 20

storage(initial 100k next 100k pctincrease

tablespace_name;

0 maxextents 100) tablespace

索引重建:

重建索引(可以同时使 存储 句和参数,不重建时也可直接使)

alter index index_name rebuild tablespace tablespace_name nologging parallel 4;

alter index index_name noparallel;

在线重建索引.可以减少加锁时间,从 开放使 DML 类型操作

alter index index_name rebuild tablespace tablespace_name nologging online;

  动拓展索引的空间

alter index index_name allocate extent;

收回未 到的空间

alter index index_name deallocate unused;

索引碎 整理

alter index index_name coalesce;

标识索引是否使 过

alter index index_name monitoring usage;

查询:

select * from v$object_usage;

取消监控

alter index index_name nomonitoring usage

索引压缩:

alter index index_name rebuild nologging online tablespace tablespace_name compress;

索引删除:

drop index index_name

索引查看:

索引相关信息

select owner,index_name,table_name,tablespace_name,index_type,degree,status

from dba_indexes;

索引列对照信息

select index_name,table_name,column_name,index_owner,table_owner

from dba_ind_columns;

索引存储信息

select index_name,pct_free,pct_increase,initial_extent,next_extent,min_extents,

max_extents from dba_indexes;

2、分区表索引

索引创建:

局部索引

```
create index index_name on table_name (column)
local
(
partition partition_name1 tablespace index_tablespace_name1,
partition partition_name2 tablespace index_tablespace_name2,
partition partition_name3 tablespace index_tablespace_name3
)
```

全局索引

```
create [unique] index index_name on table_name(column)
global partition by range(column)
(
partition partition_name1 value less than(first range value) tablespace
index_tablespace_name1,
partition partition_name2 value less than(second range value) tablespace
index_tablespace_name 2,
、、、、、、、
partition partition_nameN value less than(maxvalue) tablespace index_tablespace_nameN
)
```

```
create [unique] index index_name on table_name(column,[column2])
global partition by hash(column,[column2])
(
partition partition_name1 tablespace index_tablespace_name1,
partition partition_name2 tablespace index_tablespace_name 2,
、、、、、、、、
partition partition_nameN tablespace index_tablespace_nameN
)
```

索引重建：

```
alter index index_name rebuild tablespace tablespace_name nologging online parallel 4;
alter index index_name noparallel;
```

索引删除：

```
drop index index_name;
```

索引查看：

索引相关信息

```
select owner,index_name,table_name,tablespace_name,index_type,degree,status from
dba_indexes;
```

索引列对照信息

```
select index_name,table_name,column_name,index_owner,table_owner from
dba_ind_columns;
```

索引存储信息

```
select index_name,pct_free,pct_increase,initial_extent,next_extent,min_extents,max_extents
from dba_indexes;
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：https://d.book118.com/765034144303011220