## 1、||，distinct，desc（discribe）

1）||：字符串连接，相当于 mysql中的 +比如:'abc' || 'def' 结果为：'abcdef'；

2）distinct：去重函数。例如：`select distinct name from t_table`;

3）desc（describe）:查看表结构。例如：`DESCRIBE t_table`

说明对于表，视图，类型和同义词包含以下信息：

1. 每列的名称

2. 是否允许空值（NULL或NOT NULL）用于每列

3. 数据类型的列

4. 列的精度（以及数字列的缩放比例，如果有的话）

---

## 2、between and,IN,LIKE,IS NULL

1）between and: 例如：select * from test where num between '1' and '5';

对于orcle数据库 between and 等效 >= and <=；

2）IN：对一个查询的结果集的操作。例如：select * from 表 where 字段 in ('A','B','C');

3）LIKE：模糊查找。字符匹配操作可以使用通配符 “%” 和 “_”： %：表示任意个字符，包括零个； _：表示一个任意字符； 例如：select * from dept where DNAME like '_A%';

4）is null： 表示 字段的值的空。 注意：空值不能被索引，所以查询时有些符合条件的数据可能查不出来，count(*)中，用nvl(列名,0)处理后再查。

---

## 3、常见的单行函数，round，trunc，mod，to_number,to_char,to_date,nvl,case,decode

1）round：ROUND(a,b):四舍五入（保留a的b位小数）

```
SQL> select round(4523.1354,2) 保留两位小数,round(4523.1354,0) 保留整数,round(4523.1354,-2) 保留百位;

保留两位小数    保留整数    保留百位
--------------- ---------- ----------
4523.14       4523       4500
```

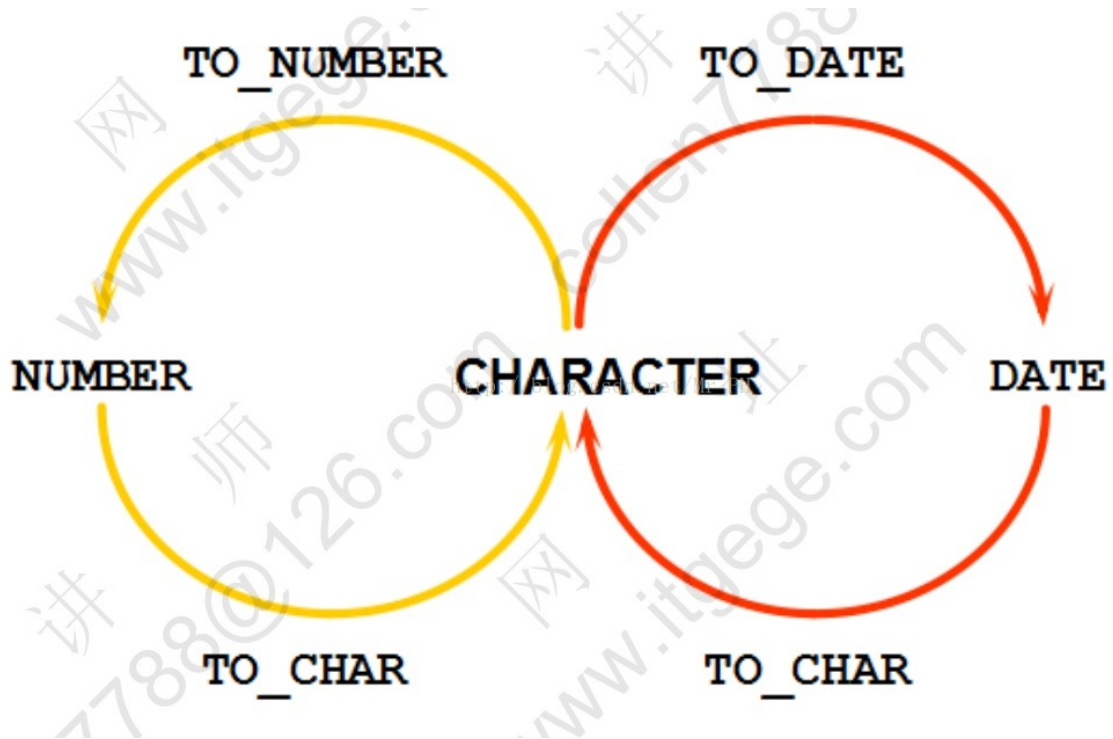
2)trunc：TRUNC(a,b) ： 截取小数

```
SQL> select trunc(4523.1354,2)截取保留两位小数,trunc(4523.1354,0) 截取整数,trunc(4523.1354,-2)
截取百位;

截取保留两位小数    截取整数    截取百位
---------------- ---------- ----------
4523.13       4523       4500
```

3）mod：MOD(a,b)：求余（a%b）

```
SQL> select mod(1520,30) from dual;

MOD(1520,30)
------------
          20
```

转换：




4）to_number：to_number('字符串')函数将字符串转成数字；

5）to_char：to_char(数值，'格式')函数将数值转成字符串；

6）to_date：to_date('字符串','格式')函数 转化为日期；

7）nvl：NVL(expr1,expr2)，第一个参数为空那么显示第二个参数的值，如果第一个参数的值不为空，则显示第一个参数本来的值。

8）case --when then --else --end：类似于if --else if----else

```
select case when id between 1 and 10 then 'low'
            when id between 20 and 30 then 'mid'
            when id between 40 and 50 then 'high'
            else 'unknow'
       end
from product;
```

9）decode：**decode**(条件,值1,返回值1

,值2,返回值2,

...值n,返回值n

,缺省值)；

类似于swich --case--

---

## 4、等值连接、自连接、左连接、右连接、左外连接、右外连接

1）等值连接：在连接条件中使用等于号(=)运算符比较被连接列的列值，其查询结果中列出被连接表中的所有列，包括其中的重复属性。 （区别于自然连接）

2）自连接：将自己表的一个镜像作为另一张表来对待

```
select e.empno,e.ename,e.mgr from emp e,emp b
where e.empno = b.mgr;
```

3)左外连接（LEFT OUTER JOIN/ LEFT JOIN）

LEFT JOIN是以左表的记录为基础的,示例中t_A可以看成左表,t_B可以看成右表,它的结果集是t_A表中的全部数据，再加上t_A表和t_B表匹配后的数据。换句话说,左表(t_A)的记录将会全部表示出来,而右表(t_B)只会显示符合搜索条件的记录。t_B表记录不足的地方均为NULL。

```
select * from t_A a left join t_B b on a.id = b.id;
或
select * from t_A a left outer join t_B b on a.id = b.id;
```

用（+）来实现， 这个+号可以这样来理解： + 表示补充，即哪个表有加号，这个表就是匹配表。如果加号写在右表，左表就是全部显示，所以是左连接。

```
Select * from t_A a,t_B b where a.id=b.id(+);
```

4)右外连接（RIGHT OUTER JOIN/RIGHT JOIN）

和LEFT JOIN的结果刚好相反,是以右表(t_B)为基础的。它的结果集是t_B表所有记录，再加上t_A和t_B匹配后的数据。 t_A表记录不足的地方均为NULL。

```
select * from t_A a right join t_B b on a.id = b.id;
或
select * from t_A a right outer join t_B b on a.id = b.id;
```

用（+）来实现， 这个+号可以这样来理解： + 表示补充，即哪个表有加号，这个表就是匹配表。如果加号写在左表，右表就是全部显示，所以是右连接。

```
Select * from t_A a,t_B b where a.id(+)=b.id;
```

---

## 5、常见的分组函数：max，min，avg，count，sum　having关键字

1）Count：统计查询结果有几条记录，如果数据库表的没有数据，`Select count(*) from dual;`

2) max，min，avg，sum

```
Select avg(comm) from emp;--平均值
Select max(comm) from emp;--最大值
Select min(comm) from emp;--最小值
Select sum(comm) from emp;--总和

--分组列中空值，可使用Nvl()函数强制分组函数处理空值
select avg(nvl(comm, 0)) from emp;
```

3) GROUP BY

1. 出现在 SELECT 列表中的字段或者出现在 order by 后面的字段，如果不是包含在分组函数中，那么该字段必须同时在 GROUP BY 子句中出现
2. 包含在 GROUP BY 子句中的字段则不必须出现在 SELECT 列表中
3. 可使用 where 字句限定查询条件
4. 可使用 Order by 子句指定排序方式

   4) having : Wheret和having子句都用来筛选数据，但是where是针对原数据进行筛选，而having子句只是针对汇总后的结果进行 。

---

## 6、insert

1、insert into 表名 values(值1，值2...) 2、insert into 表名(列名1,列名2...) values(值1,值2...) 3、insert into 表名 1 select * from 表名2

## update

1、update 表名 set 字段=值 where 2、update 表名 set 字段1=值1，字段2=值2 where。。。 3、update 表1 a set 字段=(select 字段 from 表2 b where a.关联字段=b.关联字段) where...

## delete

1、delete from 表 where。。。 2、delete from 表 3、清空表 truncate table 表名

## delete与truncate的区别：

1、DELETE可以删除满足条件的，但是TRUNCATE只能全清除 2、DELETE删除的数据可以回滚，但是TRUNCATE不能回滚 3、都来做清除的时候TRUNCATE效率高(HW high water)

## 事物（特性：ACID 隔离级别）

在数据库中事务是工作的逻辑单元，一个事务是由一个或多个完成一组的相关行为的SQL语句组成，通过事务机制确保这一组SQL语句所作的操作要么都成功执行，完成整个工作单元操作，要么一个也不执行。

SQL92标准定义了数据库事务的四个特点：

- 原子性(Atomicity)：一个事务里面所有包含的SQL语句是一个执行整体，不可分割，要么都做，要么都不做。
- 一致性(Consistency)：事务开始时，数据库中的数据是一致的，事务结束时，数据库的数据也应该是一致的。
- 隔离性(Isolation)：是指数据库允许多个并发事务同时对其中的数据进行读写和修改的能力，隔离性可以防止事务的并发执行时，由于他们的操作命令交叉执行而导致的数据不一致状态。
- 持久性 (Durability) : 是指当事务结束后，它对数据库中的影响是永久的，即便系统遇到故障的情况下，数据也不会丢失。

一组SQL语句操作要成为事务，数据库管理系统必须保证这组操作的原子性（Atomicity）、一致性（consistency）、隔离性（Isolation）和持久性（Durability），这就是ACID特性。

| 隔离级别 | 脏读 | 不可重复读 | 幻读 |
| --- | --- | --- | --- |
| Read uncommitted(读未提交) | 是 | 是 | 是 |
| Read committed（读已提交） | 否 | 是 | 是 |
| Repeatable read（可重复读） | 否 | 否 | 是 |
| Serializable（串行读） | 否 | 否 | 否 |

## 锁（共享锁(读锁，s锁) 排它锁(写锁，x锁)）

排他锁（exclusive lock，即X锁）

事务设置排它锁后，该事务单独获得此资源，另一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。

---

共享锁（share lock，即S锁）

共享锁使一个事务对特定数据库资源进行共享访问——另一事务也可对此资源进行访问或获得相同共享锁。

共享锁为事务提供高并发性，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。

---

## 7、常见的数据库对象？(表、识图、序列、索引、同义词)

**表：**

**1、创建**

`create table 表名(字段 类型,字段 类型...)` `create table 表名1 as select * from 表名2` `create table 表名1 as select * from 表名2 where 1=2`

**2、数据类型 （VARCHAR2与char区别,long、blob、clob？rowid）**

CHAR的长度是固定的，而VARCHAR2的长度是可以变化的；

---

LONG: 可变长的字符串数据，最长2G，LONG具有VARCHAR2列的特性，可以存储长文本一个表中最多一个

LONG RAW: 可变长二进制数据，最长2G 【不建议使用】

CLOB: 字符大对象Clob 用来存储单字节的字符数据；大型文本，例如XML数据。

NCLOB: 用来存储多字节的字符数据

BLOB: 用于存储二进制大对象数据；例如数码照片；

---

rownum：是伪列，是在获取查询结果集后再加上去的 （获取一条记录加一个rownum）。对符合条件的结果添加一个从1开始的序列号。

rowid是物理存在的，实际存在的一个列，是一种数据类型。 基于64为编码的18个字符来唯一标识的一条记录的物理位置的一个ID。

**3、管理(增加列，修改列，删除列)**

增加列: alter table 表名 add (字段 类型) 修改列： alter table 表名 modify (字段 类型) 删除列: alter table 表名 drop column 字段 删除表： drop table 表名

---

## 8、约束:常见的约束(not null,unique,primary key ,foreign key,check） primary key与unique?

1.—-主键约束（Primay Key Coustraint） 唯一性，非空性

2.—-唯一约束 （Unique Counstraint）唯一性，可以空，但只能有一个

3.—-检查约束 (Check Counstraint) 对该列数据的范围、格式的限制（如：年龄、性别等）

4.—-默认约束 (Default Counstraint) 该数据的默认值

5.—-外键约束 (Foreign Key Counstraint) 需要建立两表间的关系并引用主表的列

实例：

```
1.--添加主键约束（将stuNo作为主键）
alter table stuInfo
add constraint PK_stuNo primary key (stuNo)
2.--添加唯一约束（身份证号唯一，因为每个人的都不一样）
alter table stuInfo
add constraint UQ_stuID unique(stuID)
3.--添加默认约束（如果地址不填 默认为“地址不详”）
alter table stuInfo
add constraint DF_stuAddress default (‘地址不详’) for stuAddress
4.--添加检查约束 （对年龄加以限定 15-40岁之间）
alter table stuInfo
add constraint CK_stuAge check (stuAge between 15 and 40)
alter table stuInfo
add constraint CK_stuSex check (stuSex=’男’ or stuSex=’女’)
5.--添加外键约束（主表stuInfo和从表stuMarks建立关系，关联字段stuNo）
alter table stuInfo
add constraint FK_stuNo foreign key(stuNo)references stuinfo(stuNo)
```

---

## 9、视图:创建视图、删除视图

CREATE [OR REPLACE] VIEW 视图名 AS xx

问题1： 1、视图的数据是否可以修改 是可以的 2、修改了数据是否会影响到原表？ 是影响的 3、如果限制视图不能修改可以在建视图的时候指定with read only CREATE OR REPLACE VIEW v_emp AS SELECT empno,ename FROM emp WITH READ ONLY

3、删除视图 DROP VIEW 视图名 4、显示视图结构 DESC 视图名;

物化视图： 普通视图的数据和基表的数据是同一份数据 物化视图的数据和基表的数据不是同一份数据

---

## 10、序列(产生连续的值)

创建序列:`create sequence seq_test;` 使用： `select seq_test.nextval from dual;` `select seq_test.currval from dual;`

---

## 11、索引

1、索引的作用:提高查询效率 2、创建索引 create index idx_test(id) 3、删除索引 drop index idx_test;

---

## 12、union与union all、INTERSECT、minus

UNION:连接，但是有重复的会剔除 `表1 UNION 表2`

UNION ALL:连接，不去除重复的 `表1 UNION all 表2`

INTERSECT：两个集合取交集 `表1 INTERSECT表2`

minus:差集 `表1 minus表2`

---

## 13、pl/sql

PL/SQL是 Procedure Language & Structured Query Language 的缩写， 一种过程处理语言 。

---

## 14、行列转换

```
SELECT ID,NAME,
        SUM(DECODE(course,'语文',score,0)) 语文,
        SUM(DECODE(course,'数学',score,0)) 数学,
        SUM(DECODE(course,'英语',score,0)) 英语,
        SUM(DECODE(course,'历史',score,0)) 历史,
        SUM(DECODE(course,'化学',score,0)) 化学
    FROM kecheng
    GROUP BY ID ,NAME

```

---