2016/11/10

数据库

Eg：Oracle、SQLserver、DB2

T-SQL：结构化查询语言

数据库的语言类型：

1>DDL（Data definition language）：数据定义语言（创建数据库、创建用户、创建数据表、创建序列、创建索引、触发器、存储过程、函数、或事务）

2>DML（Data Manipulate language）:数据操作语言（增删改查）

3>DCL（Data control language）:数据控制语言（用于管理数据的权限（授权、取消权限））

数据库逻辑图

一个数据可以有多个不同的表空间（tablespace）

一个表空间可以有多个不同的数据表（table）

一个数据表可以有多个不同的字段（域）（field）

一个字段（域）可以有多个不同的区间

数据库用户—访问数据库的人，在数据库中，可以对数据库的各种安全参数进行设置；

系统用户：sys. ystem

System：可以直接登入

Sysman：（企业管理、管理员级别的账号）

Scott：默认密码是tiger；

数据库默认表空间

（Oracle中存在六个默认表空间）

表空间和数据文件之间的关系：

一个表空间可以有多个数据文件构成（数据文件内容、大小及位置由用户决定）；

用户需要知道的存储数据表、数据库，数据库中的一些其他对象都是存储在

Oracle：永久表空间、临时表空间、undo表空间

永久··：永久存储数据库对象；比如：数据表、视图、索引

临时~~：数据库操作过程中，中间执行的过程（操作），一旦数据库操作结束，临时表空间中的数据会销毁

Undo~~：保持事务修改的旧值（用于撤销操作，保证事务的回滚操作）

创建表空间：

Syntax：

Create [temporary] tablespace tablespace\_name tempfile|datafile ‘ \*\*\*\*.dbf’ size number M

Eg: create tablespace alex\_tablespace tempfile ‘alex.dbf’ size 100m;

修改表空间

添加表空间

Syntax:alter tablespace tablespace\_name add datafile ‘datafilename.dbf’ size numberM

Eg:alter tablespace alex\_tablespace add datafile ‘will.dbf’ size 100M

修改表空间的读写属性

Eg:alter tablespace alex\_tablespace READ ONLY;

删除表空间

Syntax：drop tablespace tablespace\_name [including content]

(Note：如果仅仅删除表空间，而不删除其中的数据文件、可以省略[including content])

数据表

(数据存储的基本单元。相当于一个类型，表中的每一行，相当于该类型事物的一个对象；

数据表本质就是对象的集合)

Eg：Student类中的字段：

ID Name age sex address

12 xiah 67 男 上海

13 xiahk 17 男 上海

……

数据表中的每一行，相当于一个对象，称之为一条记录。

数据列：相当于属性（有自己的数据类型，称之为字段/域）。

数据类型

字符类型

char(n)；—表示固定长度字符串，最大值2000

定义gender的数值char(6)（‘male’中占用6个字符 ）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| M | A | L | E |  |  |

varchar2(n)； —表示可变长度字符串，最多可以存放n个字符，最大值4000

定义gender的数值varchar2(6) （‘male’中占用4个字符 ）

|  |  |  |  |
| --- | --- | --- | --- |
| M | A | L | E |

nchar(n),nvarchar2(n)；—前面加n可以识别汉字

数值类型

Number(p,s)；（p：有效数字的位数，s：小数点后面有效位数）

Int 整数类型，number类型的子类型

Float(n)；用于存储二进制数据，二进制数据0-126

日期类型

Date：精确到秒单位，表示范围公元前4712年1月1日-公元9999年12月31日，通常通过一个toDate(‘ ’,’ ’)—获取时间信息

Timestamp：精确到小数秒；

其他类型

Blcb：存储大数据量的二进制数据

**创建表**

**(创建表“student”)**

**CREATE TABLE student (**

**id int,**

**name varchar2(200),**

**age int**

**)**

（使用:DESC student; 查看表student）

表名列名的命名规则

必须以字母开头

长度必须是1-30个字符

只能包含A-Z、a-z、0-9、\_、$和#，后两个不建议使用

不能与同一数据库用户拥有的其他对象重名

不能是Oracle服务器的保留字

表名和列名应该具有较强的表意性

不区分大小写

修改表

1>添加字段

Syntax：alter table tablename add columnname datatype;

Eg:alter table student add sorce Number(3,2);

2>修改字段

Syntax：alter table tablename modify columnname datatype;

Eg:alter table student modify sorce Number(3,3);

insert into student values(12,’xiah’,45,67.89);

3>删除字段

Syntax：alter table tablename modify columnname datatype;

4>修改字段名

**ALTER TABLE student DROP COLUMN gender;**

5>修改数据表

Syntax：rename OLDtablename to Newtablename;

数据处理

添加数据

Syntax： insert into tablesname (columnname1,columnname2,…) values(value1,value2,…)

修改数据

Syntax；update tablesname set column1=value1,…..[where column2=value2]

删除表(两种操作)

(Truncate 、Drop 都是DDL语言，操作之后数据立即更新，原始数据不会放置在rollback segment，数据无法恢复；应小心使用，删除速度Drop>Truncate>delete)

Truncate:用于清空表中的数据，而表的结构依然存在；

Syntax：truncate table tablename；

Drop：syntax（drop table tablename）(数据、结构、索引、触发器都删除)

删除数据

Delete（属于dml语句，放置在rollback seqment中，事务提交之后才会生效，如有误操作，可以进行数据恢复）

Syntax：delete from tablename [where condition]

数据的完整性和约束

数据的完整（正确+准确）

正确：数据类型正确；准确：数据的精准，要反映相应的客观现实（eg.age 0-150）

约束的种类-基于约束的级别 表级约束 vs 列级约束

表级约束:约束放置在列级约束以外

列级约束：约束数据表中的一列

约束的类型：

1>非空约束：约束该列的数值不能为空值（null）

建表时添加非空约束Eg：create table tablename（column datatype not null）；

表中修改非空约束eg：alter table tablename modify columnname [contraint];

Alter table student modify id not null;

添加字段同时，添加非空约束(前提是数据是空的)

syntax alter table tablename add columnname1 datatype [contraint];

Eg:alter table student add address nchar(30) not null;

2>唯一约束：字段的唯一性

第一种：在创建表的同时，创建唯一约束（unique）

Create table student(id int unique);

第二种：修改表时设置唯一约束

Syntax：alter table tablename add constraint constraintname unique (columnname);

Eg:alter table student add constraint idunique unique (id);

3>主键约束：确定表中每一行数据的唯一性；被设置类主键约束的字段，必须是非空和唯一的，一个数据表只能被设置一个主键；一个主键可以有多个字段组成（联合/复合主键）

第一种：在创建表的同时，创建主键约束（primary key）

Create table student(id int primary key);

第二种:在创建表时创建主键约束（表级）

Create table student(

Id int,

Name nchar(30),

Sorce int,

Contranint pk\_name primary key(id)

)

第三种：修改表时设置主键约束

Syntax：alter table tablename add constraint constraintname primary key(columnname);

Eg:alter table student add constraint idPrimarykey primary key(id);

删除主键、唯一约束(需要先查询主键约束名称)

Alter table tablename drop constraint constraintname;

查看主键约束

**SELECT constraint\_name,search\_condition FROM user\_constraints**

**WHERE table\_name=UPPER(‘student’);**

启用（禁用）主键约束（唯一约束）

Alter table tablename disable|enable constraint constraintname;

4>检查约束：保证数据的正确性，使得数据有意义；

第一种：创建表时设置

Create table student(

Id int,

Age int check(age>0 and age<150)

)

第二种：修改表时添加

Alter table tablename add constraint constraint\_name check(check-condition)

Eg:alter table student add constraint check\_age(age between 24 and 40)

5>外键约束：表示主从表主键的关联关系

第一种：创建表时设置（列级）

Syntax:create table tablename1(

Columnname1 datatype1,

Columnname2 datatype2 reference tablename2(columnnameB),)

……

)其中tablename2为主表，tablename1为从表，columnnameB主表的主键

第二种：创建表时设置（列级）

Syntax：create table tablename1(

Columnname1 datatype1,

Columnname2 datatype2,

constraint constraint\_name FOREIGN KEY(columnname) reference tablename2(columnnameB)

)

序列（sequence）:类似于SQLserver中identity；

Oracle中没有自增类型的字段需要用户自定义序列来完成自增类型数据

主要用来解决主键唯一性，确保数值的自动增长