8.1 SQL概述与数据库定义

主要考点

1、SQL的基本组成
2、SQL的数据类型
3、表的创建、修改和删除
4、索引的创建和删除
5、视图的创建和删除

SQL的基本组成

1、数据定义语言。SQL DDL提供定义关系模式和视图、删除关系和视图、修改关系模式的命令。

2、交互式数据操纵语言。SQL DML提供查询、插入、删除和修改的命令。

3、事务控制。SQL提供定义事务开始和结束的命令。

4、嵌入式SQL和动态SQL。用于嵌入到某种通用的高级语言中混合编程。其中，SQL负责操纵数据库，高级语言负责控制程序流程。

5、完整性。SQL DDL包括定义数据库中的数据必须满足的完整性约束条件的命令，对于破坏完整性约束条件的更新将被禁止。

6、权限管理。SQL DDL中包括说明对关系和视图的访问权限。

7、SQL语言中完成核心功能的9个动词：

（1）数据查询：Select

（2）数据定义：Create、Drop、Alter

（3）数据操纵：Insert、Update、Delete

（4）数据控制：Grant、Revoke

SQL的数据类型

类型	说明
char(n)	固定长度字符串，表示n个字符的固定长度字符串
varchar(n)	可变长度字符串，表示最多可以有n个字符的字符串
int	整型，也可以用integer
smallint	短整型
numeric(p,d)	定点数，p为整数位，n为小数位
real	浮点型
double precision	双精度浮点型
float(n)	n为浮点型
boolean	布尔型
date	日期型
time	时间型

表的创建、修改和删除

1、创建表

语句格式：CREATE TABLE <表名>（<列名><数据类型>[列级完整性约束条件]
[，<列名><数据类型>[列级完整性约束条件]]...
[，<表级完整性约束条件>]）；

注：[ ]表示可选，< >表示必填。

1、实体完整性约束：

（1）在列后面加 PRIMARY KEY

（2）在最后加PRIMARY KEY(属性名1，属性名2)　　//主码为属性组(两个或以上属性的组合)只能用这种方法

2、参照完整性约束：

（1）在列后面加 References 表名（属性名）

（2）在最后面加 ，有几个外码，就写几行。

Foreign Key(属性名) References 表名（属性名）
[ON DELETE[CASCADE|SET NULL]]

ON DELETE CASCADE 表示删除被参照关系的元组时，同时删除参照关系中的元组；

ON DELETE SET NULL表示删除被参照关系的元组时，将参照关系的相应属性值置为空值。

表的创建、修改和删除

3、属性值上的约束

（1）NULL：表示为空；NOT NULL表示不能为空；

（2）UNIQUE：表示取值唯一；

（3）NOT NULL UNIQUE：表示取值唯一且不为空，与属性列后面的PRIMARY KEY可互换；

（4）CHECK：限制列中值的取值范围。

如：CHECK（Sex='男' OR Sex='女'），CHECK（余额>=0），CHECK（年龄>=18 AND 年龄<=60），CHECK（离职日期 > 入职日期）

例：建立一个供应商、零件数据库。其中关系供应商S(Sno，Sname，Status，City)，关系零件P（Pno，Pname，Color，Weight，City）。该数据库要满足如下要求：

（1）供应商代码不能为空，且值是唯一的，供应商的名称也是唯一的。

（2）零件号不能为空，且值是唯一的；零件名不能为空。

（3）一个供应商可以供应多个零件，而一个零件可以由多个供应商供应。

```
Create Table S (Sno CHAR(5) NOT NULL UNIQUE,
                Sname CHAR(30) UNIQUE,
                Status CHAR(8),
                City CHAR(20),
                PRIMARY KEY(Sno));     //教材这里有错误
Create Table P (Pno CHAR(6),
                Pname CHAR(30) NOT NULL,
                Color CHAR(8),
                Weight NUMERIC(6,2),
                City CHAR(20),
                PRIMARY KEY(Pno));
Create Table SP (Sno CHAR(5),
                 Pno CHAR(6),
                 Status CHAR(8),
                 Qty NUMERIC(9),
                 PRIMARY KEY(Sno,Pno),     //如果主键为属性组合，只能用这种方法
                 FOREIGN KEY(Sno) REFERENCES S(Sno),
                 FOREIGN KEY(Pno) REFERENCES P(Pno));
```

09年下午题：

某网上书店后台数据库的部分关系模式如下：

会员（会员编号，用户名，密码，姓名，地址，邮编，电话，消费额，积分）

图书（图书编号，类型名称，图书名称，作者，出版社，出版日期，ISBN，价格）

订单（订单编号，用户名，销售额，订购日期，出货日期）

订单明细（订单明细编号，订单编号，图书编号，数量）

问题1：

下面是创建订单关系的SQL语句，订单编号唯一识别一个订单，用户名为订购图书的会员用户名，且不能为空。要求订购日期不能大于出货日期。请将空缺补充完整。

```
Create Table 订单 (
        订单编号 CHAR(6) _____（a）_____ ,
        用户名 VARCHAR(40) NOT NULL _____（b）_____ ,
        销售额 FLOAT,
        订购日期 DATE NOT NULL,
        出货日期 DATE _____（c）_____;
```

解答：（a）NOT NULL UNIQUE 或 PRIMARY KEY

（b）REFERENCES 会员（用户名）

（c）CHECK （订购日期<=出货日期）

表的创建、修改和删除

2、修改表

语句格式：ALTER TABLE <表名> [ADD <新列名><数据类型>[列级完整性约束条件]]
[DROP <完整性约束名>]
[Modify <列名><数据类型>]);

如：

```
ALTER TABLE S ADD Zap CHAR(6);                    //在表S中新增一列ZAP，该列的数据为空
ALTER TABLE S MODIFY Status INT;                  //将表S的Status属性的数据类型更改为INT
ALTER TABLE S ADD Constraint C_cno CHECK(.....)   //在表S中新增CHECK约束，取名为C_cno
```

3、删除表

语句格式：DROP TABLE <表名>

如：

```
DROP TABLE S;      //表删除后，不再是数据库模式的一部分
```

索引的创建和删除

1、索引的概念

• 数据库中的索引与书籍中的索引类似，在一本书中，利用索引可以快速查找到所需信息，无须阅读整本书。在数据库中，索引使数据库无须对整个表进行扫描，就可以在其中找到所需数据。

• 比如在字典中，我们按字母建立索引。在数据库中，索引是某个表中的一列或者若干列的值的集合，和相应的指向表中物理标识这些值的数据页的逻辑指针清单。

2、索引的作用

（1）通过创建唯一索引，可以保证数据记录的唯一性。

（2）可以大大加快数据检索速度。

（3）可以加速表与表之间的连接，这一点在实现数据的参照完整性方面有特别的意义。

（4）在使用ORDER BY和GROUP BY子句中进行检索数据时，可以显著减少查询中分组和排序的时间。

（5）使用索引可以在检索数据的过程中使用优化隐藏器，提高系统性能。

索引分为聚集索引和非聚集索引。聚集索引是指索引表索引项的顺序与表中记录的物理顺序一致的索引。

索引的创建和删除

3、建立索引：

语句格式：CREATE [UNIQUE][CLUSTER] INDEX <索引名>
ON <表名>(<列名>[<次序>][,<列名>[<次序>]]...);

- 次序：ASC(升序)或DESC（降序），默认为升序。
- UNIQUE：表明此索引的每一个索引值只对应唯一的数据记录。
- CLUSTER：表明要建立的索引是聚簇索引，索引项的顺序是与表中记录的物理顺序一致的索引组织。

如：

```
CREATE UNIQUE INDEX S_Sno on S(Sno);          //在表S的Sno列上建立索引S_Sno，默认为升序
CREATE UNIQUE INDEX P_Pno on P(Pno);          //在表P的Pno列上建立索引P_Pno，默认为升序
CREATE UNIQUE INDEX J_Jno on J(Jno);          //在表J的Jno列上建立索引J_Jno，默认为升序
CREATE UNIQUE INDEX SPJ_NO on SPJ(Sno ASC,Pno DESC,Jno ASC);
                    //在表SPJ上建立索引SPJ_NO，属性Sno按升序，Pno按降序，Jno按升序
```

4、删除索引

语句格式：DROP INDEX <索引名>

例：DROP INDEX StudentIndex;　　//删除索引StudentIndex

视图的创建和删除

1、视图的作用：自行读教材P300

2、创建视图

语句格式：CREATE VIEW 视图名(列表名)

AS SELECT 查询子句

[WITH CHECK OPTION];

- 视图的创建中，必须遵循如下规定：

（1）子查询可以是任意复杂的SELECT语句，但通常不允许含有ORDER BY子句和DISTINCT短语。

（2）WITH CHECK OPTION表示对UPDATE,INSERT,DELETE操作时保证更新、插入或删除的行满足视图定义中的谓词条件（即子查询中的条件表达式）

（3）组成视图的属性列名或者全部省略或者全部指定。如果省略属性列名，则隐含该视图由SELECT子查询目标列的主属性组成。

视图的创建和删除

例：学生关系模式S(Sno,Sname,Sage,Sex,SD,Email,Tel)，建立计算机系（CS表示计算机系）学生的视图，并要求进行修改、插入操作时保证该视图只有计算机系的学生。

```
CREATE VIEW CS_STUDENT                    //创建视图CS_STUDENT
       AS SELECT Sno,Sname,Sage,Sex       //选择学号、姓名、年龄、性别列
       FROM Student                       //从学生表中查询
       Where SD='CS'                      //选择系名等于“CS”的行
       WITH CHECK OPTION;                 //以后对该视图进行修改、插入操作时DBMS
                                            会自动加上SD='CS'的条件，
                                            保证视图中只有计算机系的学生
```

3、删除视图

语句格式：DROP VIEW 视图名

如：DROP VIEW CS_STUDENT　　//删除视图CS_STUDENT

2015年下午试题二：

【说明】

某大型集团公司的数据库的部分关系模式如下：

员工表：EMP（Eno，Ename，Age，Sex，Title），各属性分别表示员工工号、姓名、年龄、性别和职称级别，其中性别取值为“男”“女”；

公司表：COMPANY（Cno，Cname，City），各属性分别表示公司编号、名称和所在城市；

工作表：WORKS（Eno，Cno，Salary），各属性分别表示职工工号、工作的公司编号和工资。

有关关系模式的属性及相关说明如下：

（1）允许一个员工在多家公司工作，使用身份证号作为工号值。

（2）工资不能低于1500元。

【问题1】

请将下面创建工作关系的SQL语句的空缺部分补充完整，要求指定关系的主码、外码，以及工资不能低于1500元的约束。

```
CREATE TABLE WORKS(
      Eno CHAR(10) ___(a)___ ,
      Cno CHAR(4)  ___(b)___ ,
      Salary int   ___(c)___ ,
      PRIMARY KEY ___(d)___ ,
  );
```

【问题2】

（1）创建女员工信息的视图FemaleEMP，属性有Eno、Ename、Cno、Cname和Salary，请将下面SQL语句的空缺部分补充完整。

```
CREATE   ___(e)
  AS
 SELECT EMP.Eno, Ename, COMPANY.Cno, Cname, Salary
       From EMP, COMPANY, WORKS
       WHERE  ___(f)___ ;
```