数据库原理及应用

（计算机与信息学院）

实 验 报 告

2019 ----2019学年 春季学期

姓 名：

学 号：

组 别：

教 师：

**实验成绩：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |
| --- |
| 说 明  1．请在封面上按要求填写学号、姓名，及组别（如果没有分组，则不必填写）  2．按规定的时间进行实验，在下次实验前上交前次实验报告。  3．实验内容为循序渐近，每一次的实验结果都要做好保存，以备下一次实验使用。  3．按要求完成实验，上传实验代码。  4．回答思考题问题。  5．实验小结：阐述在试验过程中遇到的主要问题，以及解决方法，并总结实验收获、体会及建议  **实验四 视图及索引**  **一．实验目的与任务：**  　　要求掌握**视图的建立与维护**，**理解索引的分析方法**，索引的优缺点，掌握索引的建立与维护方法。  **二．实验内容：**  1. 建立借书、还书、查书、用户、金额统计等视图；  ①建立“借书”视图，要求显示所有用户的借书情况，要求显示借阅证号、用户姓名、单位、用户所借图书的ISBN、书号、书名、借阅时间。要求同一种用户排列在一起，同类用户按照借书证号的升序排列。  ②建立“未还书”视图：要求显示所有外借未归还的图书的ISBN、书号、借阅人名称、单位、借阅时间。  ③建立“可借图书”视图：查询所有能够外借的图书的ISBN、书号及书名  ④建立“未交罚款用户”视图：要求显示所有有超期图书记录，但未交付罚款的用户借书证号、姓名、单位及Email地址。  ⑤建立“金额统计”视图：要求显示所有用户的姓名、单位、罚款总金额，结果按照罚款总金额数的降序排列。   1. 为各表**分析**并建立索引。   USE Library;  GO  --1. 建立借书、还书、查书、用户、金额统计等视图；(借阅证号,用户姓名,单位,用户所借图书的ISBN,书号,书名,借阅时间)  /\*①建立“借书”视图，要求显示所有用户的借书情况，  要求显示借阅证号、用户姓名、单位、用户所借图书的ISBN、书号、书名、借阅时间。  要求同一种用户排列在一起，同类用户按照借书证号的升序排列。\*/  --ALTER TABLE LOAN ADD 归还日期 DATE;  --UPDATE LOAN SET LOAN.归还日期=GETDATE();  --SELECT \* FROM LOAN UNION ALL SELECT \* FROM LOANHIST;  GO  --在创建视图之前需要先执行update语句以更新系统日期  CREATE VIEW 借书  AS SELECT TOP 99.9999 PERCENT \_USER.借阅证号,姓名 AS 用户姓名,单位,BOOKS.ISBN AS 用户所借图书的ISBN,LOANX.书号,BOOKINFO.书名,DATEDIFF(DAY,LOANX.借阅日期,LOANX.归还日期) AS 借阅时间  FROM \_USER LEFT OUTER JOIN (SELECT \* FROM LOAN UNION ALL SELECT \* FROM LOANHIST)AS LOANX ON LOANX.借阅证号 =\_USER .借阅证号,BOOKS,BOOKINFO  WHERE LOANX.书号 = BOOKS.书号 AND BOOKS.ISBN = BOOKINFO.ISBN  GROUP BY \_USER.借阅证号,姓名,单位,BOOKS.ISBN,LOANX.书号,BOOKINFO.书名,LOANX.借阅日期,LOANX.归还日期,\_USER.分类  ORDER BY \_USER.借阅证号  WITH CHECK OPTION  GO  --②建立“未还书”视图：  --要求显示所有外借未归还的图书的ISBN、书号、借阅人名称、单位、借阅时间。  GO  CREATE VIEW 未还书  AS SELECT \_USER.姓名,\_USER.单位,BOOKS.ISBN,LOAN.书号,DATEDIFF(DAY,LOAN.借阅日期,GETDATE()) AS 借阅时间 FROM \_USER,LOAN,BOOKS  WHERE LOAN.借阅证号 = \_USER.借阅证号 AND LOAN.书号 = BOOKS.书号  GROUP BY LOAN.借阅证号,\_USER.姓名,\_USER.单位,BOOKS.ISBN,LOAN.书号,LOAN.借阅日期  WITH CHECK OPTION  GO  --③建立“可借图书”视图：查询所有能够外借的图书的ISBN、书号及书名  CREATE VIEW 可借图书  AS SELECT BOOKS.ISBN,书号,书名 FROM BOOKS,BOOKINFO  WHERE BOOKS.状态 = '0' AND BOOKS.ISBN = BOOKINFO.ISBN  WITH CHECK OPTION  GO  --DROP VIEW IF EXISTS 借书;  --④建立“未交罚款用户”视图：要求显示所有有超期图书记录，  --但未交付罚款的用户借书证号、姓名、单位及Email地址。(待定)  CREATE VIEW 未交罚款用户  AS SELECT LOAN.借阅证号,\_USER.姓名,\_USER.单位 ,\_USER.E\_mail  FROM \_USER,CLASS\_USER,LOAN  WHERE LOAN.借阅证号 = \_USER.借阅证号 AND \_USER.分类 = CLASS\_USER.编号  AND DATEDIFF(DAY,LOAN.借阅日期,GETDATE())>CLASS\_USER.借阅期限  AND LOAN.借阅证号 NOT IN(SELECT 借阅证号 FROM MONEY WHERE 原因='过期还书罚款')  WITH CHECK OPTION  GO  --⑤建立“金额统计”视图：要求显示所有用户的姓名、单位、罚款总金额，  --结果按照罚款总金额数的降序排列。  --DROP VIEW IF EXISTS 金额统计  CREATE VIEW 金额统计  AS SELECT TOP 99 PERCENT SUM(金额) AS 罚款总金额,姓名,单位  FROM \_USER,MONEY  WHERE MONEY.借阅证号 = \_USER.借阅证号  GROUP BY 姓名,单位,MONEY.借阅证号  ORDER BY 罚款总金额 DESC  WITH CHECK OPTION  --2.为各表分析并建立索引。  CREATE UNIQUE INDEX Userno ON \_USER(借阅证号);/\*主键\*/  CREATE INDEX Username ON \_USER(姓名);/\*常用于where子句\*/  CREATE UNIQUE INDEX User\_no ON \_USER(分类); /\*外键\*/  CREATE UNIQUE INDEX Bookinfoisbn ON BOOKINFO(ISBN);/\*主键\*/  CREATE INDEX Bookinfomoney ON BOOKINFO(价格);/\*常用于order by子句中\*/  CREATE UNIQUE INDEX Bookno ON BOOKS(书号);/\*主键\*/  CREATE INDEX Bookisbn ON BOOKS(ISBN);/\*外键\*/  CREATE UNIQUE INDEX Classuserclass ON CLASS\_USER(类别);/\*主键\*/  CREATE UNIQUE INDEX Loanno ON LOAN(借阅证号);/\*外键\*/  CREATE UNIQUE INDEX Loanbno ON LOAN(书号);/\*外键\*/  CREATE UNIQUE INDEX Loanhistno ON LOANHIST(借阅证号);/\*外键\*/  CREATE UNIQUE INDEX Loanhistbno ON LOAN(书号);/\*外键\*/  CREATE UNIQUE INDEX Moneyno ON MONEY(借阅证号);/\*外键\*/  CREATE UNIQUE INDEX Moneybno ON MONEY(书号);/\*外键\*/  CREATE INDEX \_Money ON MONEY(金额);/\*常用于order by子句中\*/  CREATE UNIQUE INDEX Reservationisbn ON RESERVATION(ISBN);/\*外键\*/  CREATE UNIQUE INDEX Reservationo ON RESERVATION(借书证号);/\*外键\*/ |
| **三．思考题：**  1.哪些情况下要明确指定视图的列名称？   1. 某个目标列名有聚集函数或者表达式； 2. 多表连接时存在同名作为视图的字段； 3. 管理员要为视图取新名字。   2.站在数据库管理员的角度，你认为上述Library数据库应该创建哪些常用视图？  建立用户借还书记录、查书、书籍借还、金额统计、书籍被借频次等视图；  3.建立索引的优缺点是什么？如何分析确定一个表上应该建立哪些索引？  (1)创建索引可以大大提高系统的性能，**优点**:  第一，通过创建唯一性索引，可以保证数据库表中每一行数据的唯一性。 　 第二，可以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。 　 第三，可以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。 　 第四，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。 　 第五，通过使用索引，可以在查询的过程中，使用优化隐藏器，提高系统的性能。  (2)增加索引也有许多不利的一个方面, **缺点**：  第一，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。 第二，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。 第三，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，这样就降低了数据的维护速度。  (3)一般来说，应该在这些列上创建索引，  1)经常需要搜索的列上，可以加快搜索的速度；   1. 在作为主键的列上，强制该列的唯一性和组织表中数据的排列结构； 2. 在经常用在连接的列上，这些列主要是一些外键，可以加快连接的速度； 3. 在经常需要根据范围进行搜索的列上创建索引，因为索引已经排序，其指定的范围是连续的； 4. 在经常需要排序的列上创建索引，因为索引已经排序，这样查询可以利用索引的排序，加快排序查询时间； 5. 在经常使用在WHERE子句中的列上面创建索引，加快条件的判断速度。   2)对于有些列不应该创建索引。一般来说，不应该创建索引的的这些列具有下列特点：   1. 对于那些在查询中很少使用或者参考的列不应该创建索引。这是因为，既然这些列很少使用到，因此有索引或者无索引，并不能提高查询速度。相反，由于增加了索引，反而降低了系统的维护速度和增大了空间需求。 2. 对于那些只有很少数据值的列也不应该增加索引。这是因为，由于这些列的取值很少，例如人事表的性别列，在查询的结果中，结果集的数据行占了表中数据行的很大比例，即需要在表中搜索的数据行的比例很大。增加索引，   并不能明显加快检索速度。   1. 对于那些定义为text, image和bit数据类型的列不应该增加索引。这是因为，这些列的数据量要么相当大，要么取值很少。 2. 当修改性能远远大于检索性能时，不应该创建索引。这是因为，修改性能和检索性能是互相矛盾的。   　　当增加索引时，会提高检索性能，但是会降低修改性能。当减少索引时，会提高修改性能，降低检索性能。因此，当修改性能远远大于检索性能时，不应该创建索引。   1. 上述Library数据库的表中，你在哪些表的哪些字段上创建了什么类型的索引？为什么？  |  |  |  |  | | --- | --- | --- | --- | | 表名 | 字段名 | 索引类型 | 原因 | | \_USER | 借阅证号 | UNIQUE | 主键 | |  | 姓名 |  | 常用于where子句 | |  | 分类 | UNIQUE | 外键 | | BOOKINFO | ISBN | UNIQUE | 主键 | |  | 价格 |  | 常用于order by子句中 | | BOOKS | 书号 | UNIQUE | 主键 | |  | ISBN | UNIQUE | 外键 | | CLASS\_USER | 类别 | UNIQUE | 主键 | | LOAN | 借阅证号 | UNIQUE | 外键 | |  | 书号 | UNIQUE | 外键 | | LOANHIST | 借阅证号 | UNIQUE | 外键 | |  | 书号 | UNIQUE | 外键 | | MONEY | 借阅证号 | UNIQUE | 外键 | |  | 书号 | UNIQUE | 外键 | |  | 金额 |  | 常用于order by子句中 | | RESERVATION | ISBN | UNIQUE | 外键 | |  | 借书证号 | UNIQUE | 外键 | | |
|  | |