|  |
| --- |
| 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  **实验三 SQL查询设计II**  **一、目的与任务：**  　要求熟练掌握SQL查询语言的联接查询、嵌套子查询、相关子查询及数据更新命令。  **二、实验内容：**   　　在实验一所创建的Library数据库中实现如下查询：   1. 内联接查询在管理学院资料室的数据库类书的信息。 2. 左外联接查询分类为3的用户的姓名、单位、及借阅情况。 3. 使用子查询查询与借阅证号为“J00016”的用户在同一单位的所有用户的借阅证号和姓名。 4. 使用子查询查询所有借书预约成功的用户的姓名和E\_mail，以便通知他们。 5. 使用子查询查询类别为“教师”的用户的借书情况。 6. 计算相关子查询查询借阅数量大于3本的用户的借阅证号、姓名、单位。 7. 查询所有曾经借过书号为“A0450049”这本书的所有用户的借阅证号和姓名。 8. 查询所有借过书的用户借阅证号。 9. 查询现在正借有书的用户但以前没有借过书的用户的借阅证号。 10. 查询当前所有借书信息，并将查询结果保存到表TE中。 11. 新建一个表，包括ISBN号和借阅次数两个字段，通过查询将每类书的ISBN号和借阅次数添加到这个表中。 12. 在USER表中添加一个字段金额，并对每个用户的交费总额进行修改。   USE Library;  ALTER TABLE LOAN ADD 归还日期 DATE;  --1.内联接查询在管理学院资料室的数据库类书的信息。(内连接即是自然连接)  SELECT BOOKINFO.ISBN,书号,书名,作者,出版社,价格,语言,页数,位置,状态,借阅对象 FROM BOOKINFO,BOOKS WHERE BOOKS.位置 ='管理学院资料室' AND BOOKS.ISBN=BOOKINFO.ISBN;  --2.左外联接查询分类为3的用户的姓名、单位、及借阅情况。  SELECT \_USER.借阅证号,姓名,单位,书号,借阅日期,分类 FROM \_USER LEFT OUTER JOIN LOAN ON (\_USER.借阅证号 = LOAN.借阅证号) WHERE 分类='3';  --3.使用子查询查询与借阅证号为“J00016”的用户在同一单位的所有用户的借阅证号和姓名。  SELECT 借阅证号, 姓名 FROM \_USER WHERE 单位 IN (SELECT 单位 FROM \_USER WHERE 借阅证号 = 'J00016');  --4.使用子查询查询所有借书预约成功的用户的姓名和E\_mail，以便通知他们。  SELECT 姓名,E\_mail FROM \_USER WHERE 借阅证号 IN (SELECT 借书证号 FROM RESERVATION WHERE 状态='T');  --5.使用子查询查询类别为“教师”的用户的借书情况。  SELECT \* FROM LOAN WHERE 借阅证号 IN(SELECT 借阅证号 FROM \_USER WHERE 分类 IN (SELECT 编号 FROM CLASS\_USER WHERE 类别 = '教师'));  --6.计算相关子查询查询借阅数量大于3本的用户的借阅证号、姓名、单位。  SELECT 借阅证号,姓名, 单位 FROM \_USER x WHERE 借阅证号 = (SELECT COUNT(书号) FROM LOAN y WHERE x.借阅证号 = y.借阅证号);  --7.查询所有曾经借过书号为“A04500049”这本书的所有用户的借阅证号和姓名。  SELECT LOANHIST.借阅证号, 姓名 FROM \_USER,BOOKS,LOANHIST  WHERE BOOKS.书号 = 'A04500049' AND LOANHIST.借阅证号 = \_USER.借阅证号 AND LOANHIST.书号 = BOOKS.书号  --8.查询所有借过书的用户借阅证号。  SELECT DISTINCT \_USER.借阅证号 FROM \_USER,LOAN,LOANHIST WHERE \_USER.借阅证号 = LOAN.借阅证号 OR \_USER.借阅证号 = LOANHIST.借阅证号  --9.查询现在正借有书的用户但以前没有借过书的用户的借阅证号。  SELECT DISTINCT \_USER.借阅证号 FROM \_USER,LOAN,LOANHIST WHERE \_USER.借阅证号 = LOAN.借阅证号 AND \_USER.借阅证号 != LOANHIST.借阅证号  --10.查询当前所有借书信息，并将查询结果保存到表TE中。  SELECT \* INTO TE FROM LOAN;  --11.新建一个表，包括ISBN号和借阅次数两个字段，通过查询将每类书的ISBN号和借阅次数添加到这个表中。(待定)  SELECT ISBN,COUNT(书号) AS 借阅次数 INTO 书籍借阅信息  FROM (SELECT LOAN.\*,BOOKS.ISBN FROM (LOAN LEFT OUTER JOIN BOOKS ON LOAN.书号 = BOOKS.书号)  UNION ALL SELECT LOANHIST.\*,BOOKS.ISBN FROM (LOANHIST LEFT OUTER JOIN BOOKS ON LOANHIST.书号 = BOOKS.书号))AS X  GROUP BY ISBN;  --12.在USER表中添加一个字段金额，并对每个用户的交费总额进行修改。(待定)  ALTER TABLE \_USER ADD 金额 REAL;  ALTER TABLE \_USER ADD CONSTRAINT DF\_\_USER\_金额 DEFAULT (0) FOR 金额;  UPDATE \_USER SET 金额 =  (SELECT SUM(金额) FROM MONEY GROUP BY 借阅证号 HAVING MONEY.借阅证号 = \_USER.借阅证号); |
| **三、思考题**：   1. 内联接、左外联接、右外联接、全联接的结果有什么区别？ 2. 内连接：仅将两个表中满足连接条件的行组合起来作为结果集； 3. 左外连接：在内连接的基础上，还包含左表中所有不符合条件的数据行，并在其中的右表列填写NULL，当在内连接查询中加入条件是，无论是将它加入到join子句，还是加入到where子句，其效果是完全一样的，但对于外连接情况就不同了。当把条件加入到 join子句时，SQL Server、Informix会返回外连接表的全部行，然后使用指定的条件返回第二个表的行。如果将条件放到where子句 中，SQL Server将会首先进行连接操作，然后使用where子句对连接后的行进行筛选。 4. 右外连接：在内连接的基础上，还包含两个表中所有不符合条件的数据行，并在其中的左表、和右表列填写NULL 5. 全连接：将两个表的所有行进行组合，连接后的行数为两个表的乘积数（笛卡儿积）。 6. 嵌套子查询和相关子查询的执行过程有什么区别？ 7. 嵌套子查询的执行不依赖与外部的查询。   执行过程：   1. 执行子查询，其结果不被显示，而是传递给外部查询，作为外部查询的条件使用。 2. 执行外部查询，并显示整个结果。 3. 相关子查询的执行依赖于外部查询。多数情况下是子查询的WHERE子句中引用了外部查询的表。   执行过程：   1. 从外层查询中取出一个元组，将元组相关列的值传给内层查询。 2. 执行内层查询，得到子查询操作的值。 3. 外查询根据子查询返回的结果或结果集得到满足条件的行。 4. 然后外层查询取出下一个元组重复做步骤1-3，直到外层的元组全部处理完毕。      1. 有哪些方法可以提高SQL的查询效率？ 2. 对查询进行优化，应尽量避免全表扫描，首先应考虑在 where 及 order by 涉及的列上建立索引。 3. 应尽量避免在 where 子句中对字段进行 null 值判断，否则将导致引擎放弃使用索引而进行全表扫描，可以在num上设置默认值0，确保表中num列没有null值，然后这样查询： 4. select id from t where num=0 5. 应尽量避免在 where 子句中使用!=或<>操作符，否则将引擎放弃使用索引而进行全表扫描。 6. 应尽量避免在 where 子句中使用 or 来连接条件，否则将导致引擎放弃使用索引而进行全表扫描。 7. in 和 not in 也要慎用，否则会导致全表扫描，对于连续的数值，能用 between 就不要用 in 。 8. 如果在 where 子句中使用参数，也会导致全表扫描。因为SQL只有在运行时才会解析局部变量，但优化程序不能将访问计划的选择推迟到运行时；它必须在编译时进行选择。然而，如果在编译时建立访问计划，变量的值还是未知的，因而无法作为索引选择的输入项。 9. 应尽量避免在 where 子句中对字段进行表达式操作，这将导致引擎放弃使用索引而进行全表扫描。 10. 应尽量避免在where子句中对字段进行函数操作，这将导致引擎放弃使用索引而进行全表扫描。 11. 不要在 where 子句中的“=”左边进行函数、算术运算或其他表达式运算，否则系统将可能无法正确使用索引。 12. 在使用索引字段作为条件时，如果该索引是复合索引，那么必须使用到该索引中的第一个字段作为条件时才能保证系统使用该索引，否则该索引将不会被使用，并且应尽可能的让字段顺序与索引顺序相一致。 13. 不要写一些没有意义的查询，如需要生成一个空表结构： 14. 很多时候用 exists 代替 in 是一个好的选择： 15. 并不是所有索引对查询都有效，SQL是根据表中数据来进行查询优化的，当索引列有大量数据重复时，SQL查询可能不会去利用索引，如一表中有字段sex，male、female几乎各一半，那么即使在sex上建了索引也对查询效率起不了作用。 16. 索引并不是越多越好，索引固然可以提高相应的 select 的效率，但同时也降低了 insert 及 update 的效率，因为 insert 或 update 时有可能会重建索引，所以怎样建索引需要慎重考虑，视具体情况而定。一个表的索引数最好不要超过6个，若太多则应考虑一些不常使用到的列上建的索引是否有必要。 17. 应尽可能的避免更新 clustered 索引数据列，因为 clustered 索引数据列的顺序就是表记录的物理存储顺序，一旦该列值改变将导致整个表记录的顺序的调整，会耗费相当大的资源。若应用系统需要频繁更新 clustered 索引数据列，那么需要考虑是否应将该索引建为 clustered 索引。 18. 尽量使用数字型字段，若只含数值信息的字段尽量不要设计为字符型，这会降低查询和连接的性能，并会增加存储开销。这是因为引擎在处理查询和连接时会逐个比较字符串中每一个字符，而对于数字型而言只需要比较一次就够了。 19. 尽可能的使用 varchar/nvarchar 代替 char/nchar ，因为首先变长字段存储空间小，可以节省存储空间，其次对于查询来说，在一个相对较小的字段内搜索效率显然要高些。 20. 任何地方都不要使用 select \* from t ，用具体的字段列表代替“\*”，不要返回用不到的任何字段。 21. 尽量使用表变量来代替临时表。如果表变量包含大量数据，请注意索引非常有限（只有主键索引）。 22. 避免频繁创建和删除临时表，以减少系统表资源的消耗。 23. 临时表并不是不可使用，适当地使用它们可以使某些例程更有效，例如，当需要重复引用大型表或常用表中的某个数据集时。但是，对于一次性事件，最好使用导出表。 24. 在新建临时表时，如果一次性插入数据量很大，那么可以使用 select into 代替 create table，避免造成大量 log ，以提高速度；如果数据量不大，为了缓和系统表的资源，应先create table，然后insert。 25. 如果使用到了临时表，在存储过程的最后务必将所有的临时表显式删除，先 truncate table ，然后 drop table ，这样可以避免系统表的较长时间锁定。 26. 尽量避免使用游标，因为游标的效率较差，如果游标操作的数据超过1万行，那么就应该考虑改写。 27. 使用基于游标的方法或临时表方法之前，应先寻找基于集的解决方案来解决问题，基于集的方法通常更有效。 28. 与临时表一样，游标并不是不可使用。对小型数据集使用 FAST\_FORWARD 游标通常要优于其他逐行处理方法，尤其是在必须引用几个表才能获得所需的数据时。在结果集中包括“合计”的例程通常要比使用游标执行的速度快。如果开发时间允许，基于游标的方法和基于集的方法都可以尝试一下，看哪一种方法的效果更好。 29. 在所有的存储过程和触发器的开始处设置 SET NOCOUNT ON ，在结束时设置 SET NOCOUNT OFF 。无需在执行存储过程和触发器的每个语句后向客户端发送 DONE\_IN\_PROC 消息。 30. 尽量避免大事务操作，提高系统并发能力。 31. 尽量避免向客户端返回大数据量，若数据量过大，应该考虑相应需求是否合理 32. 避免将字段设为“允许为空” 33. 数据表设计要规范 34. 深入分析数据操作所要对数据库进行的操作 35. 尽量不要使用临时表 36. 多多使用事务 37. 尽量不要使用游标 38. 避免死锁 39. 要注意读写锁的使用 40. 不要打开大的数据集 41. 不要使用服务器端游标 42. 在程序编码时使用大数据量的数据库 43. 不要给“性别”列创建索引 44. 注意超时问题 45. 不要使用Select \* 46. 在细节表中插入纪录时，不要在主表执行Select MAX(ID) 47. 尽量不要使用TEXT数据类型 48. 使用参数查询 49. 不要使用Insert导入大批的数据 50. 学会分析查询 51. 使用参照完整性 52. 用INNER JOIN 和LEFT JOIN代替Where |
|  |