数据库技术及应用实验报告

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 姓名 | |  | 学号 |  | 班级 |  |
| 指导教师 | | 张千帆 | 成绩 |  | | |
| 实验目的 | 1. 熟练掌握使用 SQL Server 2012 完成数据库的创建、删除；数据表的建立、删除；表结构的修改；数据的插入、删除、修改与查询；视图的建立和查询。 2. 加深对表的实体完整性、参照完整性和用户自定义完整性的理解。 | | | | | |
| 实验数据 | 注：本栏按照“数据库技术及应用实验”中的实验内容，逐一写出题目、代码和 实验结果截图。  实验十：综合实验  [实验内容]  表 1 为某长途汽车站发车时刻表，其中列出了各条线路的时间、里程、车型、全程票价和需用时间，出发时间中的“\*”表示每隔 1 小时安排一班车次。车站一共有 20 名司机，30 名乘务人员和 2 名售票人员。每种车型分别有 3 台车。顾客可以查询某线路的信息进行网上车票预订或到售票窗口现购。车站管理人员每周六给司机和乘务人员安排下一周的出车任务。排班原则为一台汽车配一名司机和一名乘务人员。  任务要求：   1. 分析和设计该车站数据库系统的 E-R 模型。   图1-1  ER图version3   1. 根据 E-R 模型，设计该数据库系统的关系模型。   图1-2  实验10关系模型   1. 使用 SQL 语句，依据该关系模型建立各数据表以及索引。   代码：  create table Employee(  eId int identity(1,1),  eName varchar(20),  eSex varchar(2),  eAge tinyint,  ePosition varchar(40),  ePhone varchar(11),  constraint eSex check (eSex = '男' or eSex = '女'),  constraint pk\_employee primary key(eId)  )  go  create table Vehicle(  vId int identity(1,1),  vType varchar(20),  vLoadcount tinyint,  vNo varchar(20),  constraint pk\_Vehicle primary key(vId)  )  go  --车辆与员工对应,因为一辆车一个司机一个乘务人员  create table Empveh(  evId int identity(1,1),  vId int,  eId int,  constraint fk\_empveh1 foreign key(vId) references Vehicle(vId),  constraint fk\_empveh2 foreign key(eId) references Employee(eId),  constraint pk\_Empveh primary key(evId)  )  go  create table Routeinfo(  rId int identity(1,1),  vId int,  rNo varchar(20),  rDeparture varchar(30),  rDetination varchar(30),  rMileague smallint,  rStarttime varchar(20),  rTimecost smallint,  rDeparturedate varchar(11),  rNote varchar(50),  constraint pk\_routeinfo primary key(rId),  constraint fk\_routeinfo foreign key(vId) references Vehicle(vId)  )  go  create table Ticket(  tId int identity(1,1),  rId int,  tTotalvotes tinyint,  tRemainvotes tinyint,  tPrice float,  constraint pk\_ticket primary key(tId)  )  Go  索引：  create nonclustered index routeinfo\_vId on routeinfo(vId)  **实验结果截图**：  图1-3  0301   1. 为该车站中的乘客，司机、乘务人员、售票员和管理员建立角色和使用数据库的权限。   **代码**：  exec sp\_addrole 'passenger\_role'--乘客  exec sp\_addrole 'driver\_role'--司机  exec sp\_addrole 'crew\_role'--乘务人员  exec sp\_addrole 'saleman\_role'--售票员  exec sp\_addrole 'manage\_role'--管理员  GRANT SELECT ON Ticket TO passenger\_role--乘客 只有查询票的信息 即ticket  GRANT SELECT ON Routeinfo TO driver\_role--司机 只有查询自己运的路线信息  GRANT SELECT ON Routeinfo TO crew\_role--乘务人员 只有查询自己运的路线信息  GRANT SELECT,update,delete,insert ON Ticket TO saleman\_role--售票员 ，可以修改 增加 查询票 余数 ，总数，票价  GRANT SELECT,update,delete,insert ON Routeinfo TO manage\_role--管理员，可以对发车信息表操作，因为要增加发车信息 以及修改 这些  **运行结果截图**：  图1-4  0402   1. 根据数据库的不同用户使用数据目的不同，为各类角色定义相应的视图。   **代码**：  GO  CREATE VIEW passenger\_routeticket  AS  SELECT t.tPrice as 票价, t.tRemainvotes as 剩余坐位, v.vType as 车型,v.vNo as 车牌号,r.rStarttime as 发车时间  FROM Routeinfo r  LEFT JOIN Ticket t ON r.rId = t.rId  INNER JOIN Vehicle v ON r.vId = v.vId  WHERE rDeparturedate = CONVERT(varchar(10), getdate(), 120)  GO  GRANT SELECT ON passenger\_routeticket TO passenger\_role--赋予权限  --司机和乘务人员一样，看今天被安排哪个车 运哪条线  GO  CREATE VIEW drivercrew\_routvehc\_view  AS  SELECT emp.eName, ve.vNo, rout.rNo, rout.rDeparture, rout.rDetination  , rout.rMileague, rout.rStarttime, rout.rTimecost, rout.rDeparturedate  FROM Employee emp  INNER JOIN Empveh empve ON empve.eId = emp.eId  INNER JOIN Vehicle ve ON empve.vId = ve.vId  INNER JOIN Routeinfo rout ON rout.vId = ve.vId  WHERE rDeparturedate = CONVERT(varchar(10), getdate(), 120)  GO  GRANT SELECT ON drivercrew\_routvehc\_view TO driver\_role  GRANT SELECT ON drivercrew\_routvehc\_view TO crew\_role  --售票员可以看到发车信息,哪辆车，车的核载人数 来修改增加ticket表的数据，但为了防止对发车信息表的操作，需要设置一个视图  GO  CREATE VIEW salesman\_routeinfo\_view  AS  SELECT ve.vLoadcount,rout.\*  FROM Routeinfo rout  inner join Vehicle ve  on ve.vId = rout.vId  GO  GRANT SELECT ON salesman\_routeinfo\_view TO saleman\_role  --管理人员 安排下一周乘务人员和司机时间表，所以应该得看到下一周已经安排的路线信息 ，司机，乘务人员，名字，哪辆车  GO  CREATE VIEW manage\_routvehc\_view  AS  SELECT emp.eName,emp.ePosition, ve.vNo, rout.rNo, rout.rDeparture, rout.rDetination  , rout.rMileague, rout.rStarttime, rout.rTimecost, rout.rDeparturedate  FROM Employee emp  INNER JOIN Empveh empve ON empve.eId = emp.eId  INNER JOIN Vehicle ve ON empve.vId = ve.vId  INNER JOIN Routeinfo rout ON rout.vId = ve.vId  WHERE convert(datetime, rout.rDeparturedate) > DATEADD(wk, DATEDIFF(wk, 0, DATEADD(dd, 7, getdate())), 0)  AND convert(datetime, rout.rDeparturedate) < DATEADD(wk, DATEDIFF(wk, 0, DATEADD(dd, 14, getdate())), 0)  GO  GRANT SELECT ON manage\_routvehc\_view TO manage\_role   1. 请帮助某司机查询本周任务安排表。   **代码：**  SELECT emp.eName, ve.vType, ve.vNo, rt.rNo, rt.rDeparture  , rt.rDetination, rt.rMileague, rt.rStarttime, rt.rTimecost, rt.rDeparturedate  FROM Employee emp  INNER JOIN Empveh ev ON emp.eId = ev.eId  INNER JOIN Vehicle ve ON ev.vId = ve.vId  INNER JOIN Routeinfo rt ON ve.vId = rt.vId  WHERE convert(datetime, rt.rDeparturedate) > DATEADD(wk, DATEDIFF(wk, 0, getdate()), 0)  AND convert(datetime, rt.rDeparturedate) < DATEADD(wk, DATEDIFF(wk, 0, DATEADD(dd, 7, getdate())), 0)  AND emp.eName = '刘一'  **实验结果截图：**  图1-5  0601   1. 请从数据库中查询当天武汉到景德镇的票价、车型以及剩余座位数。   **代码：**  SELECT t.tPrice as 票价, t.tRemainvotes as 剩余坐位, v.vType as 车型,v.vNo as 车牌号,r.rStarttime as 发车时间,  r.rDeparture,r.rDetination  FROM Routeinfo r  LEFT JOIN Ticket t ON r.rId = t.rId  INNER JOIN Vehicle v ON r.vId = v.vId  WHERE rDeparturedate = CONVERT(varchar(10), getdate(), 120)  AND rDeparture = '武汉'  AND rDetination = '景德镇'  **实验结果截图**：  图1-6  0701   1. 有多个售票点同时卖票，请考虑如何保证数据的一致性。   **答：**售票即是修改Ticket表中剩余票数属性的值，就可以当做修改值，把问题变成有多个修改语句同时进行如何保证数据的一致性？把每一个修改语句变为一个事务，事务里面用数据库的排它锁（X），在某个售票点售票时在修改剩余车票前给表加上排它锁，这样其它事务就需要等待，等待先拥有排它锁的事务结束释放得到排它锁的功能即可保证数据一致性。  代码：  begin tran  declare @salecount int  declare @remain int  select \* from Ticket (tablockx) where rId = 1  update Ticket set tRemainvotes = tRemainvotes - @salecount where rId = 1  select @remain = tRemainvotes from Ticket where rId = 1  if @remain > 0  begin  commit tran  end  begin  rollback tran  end  9）统计车型为大宇的汽车在本周内运营的总金额和乘客总人数。  **代码**：  --先修改剩余票，假装售了 2 张票  update Ticket set tRemainvotes = 38 where rId = 1  update Ticket set tRemainvotes = 38 where rId = 2  update Ticket set tRemainvotes = 18 where rId =6  update Ticket set tRemainvotes = 18 where rId =7  --统计运营总金额之前一览  select tic.tId,tic.rId,ve.vType as 车型,tTotalvotes as 总票数,tic.tRemainvotes as 剩余票,tic.tprice as 价格  from Ticket tic  inner join Routeinfo rt  on rt.rId = tic.rId  inner join Vehicle ve  on ve.vId = rt.vId  WHERE convert(datetime, rt.rDeparturedate) > DATEADD(wk, DATEDIFF(wk, 0, getdate()), 0)  AND convert(datetime, rt.rDeparturedate) < DATEADD(wk, DATEDIFF(wk, 0, DATEADD(dd, 7, getdate())), 0)  AND ve.vType = '大宇'  SELECT SUM(ti.tTotalvotes - ti.tRemainvotes) AS 乘客总人数  , SUM((ti.tTotalvotes - ti.tRemainvotes) \* ti.tPrice) AS 运营总金额  FROM Routeinfo rt  INNER JOIN Ticket ti ON ti.rId = rt.rId  INNER JOIN Vehicle ve ON rt.vId = ve.vId  WHERE convert(datetime, rt.rDeparturedate) > DATEADD(wk, DATEDIFF(wk, 0, getdate()), 0)  AND convert(datetime, rt.rDeparturedate) < DATEADD(wk, DATEDIFF(wk, 0, DATEADD(dd, 7, getdate())), 0)  AND ve.vType = '大宇'  GROUP BY vType  **实验结果截图：**  图1-7  0901  0902  **实验数据**  插入值代码：  insert into employee  values('刘一','男',23,'司机',11111111111),  ('陈二','男',24,'司机',11111111111),  ('张三 ','男',24,'司机',11111111111),  ('小红 ','女',24,'乘务人员',11111111111),  ('小花 ','女',21,'乘务人员',11111111111),  ('赵六','女',23,'乘务人员',11111111111),  ('孙七','男',24,'售票人员',11111111111),  ('周八 ','男',24,'售票人员',11111111111)  insert into Vehicle  values('大宇',40,'粤B549SB'),('大宇',40,'粤BQQ001'),  ('大宇',40,'粤B22222'),('现代',20,'粤B45678'),  ('现代',20,'粤BZZZ22'),('现代',20,'粤BGZ567')  insert into empveh  values(1,1),(1,4),(2,2),(2,5),(3,3),(3,6)  insert into Routeinfo  values(1,'Z840','武汉','景德镇',380,'08:40',4,CONVERT(varchar(10), getdate(), 120 ),NULL),  (2,'K640','武汉','天门',380,'09:40',3,CONVERT(varchar(10), getdate(), 120 ),NULL),  (4,'K140','武汉','孝感',74,'18:50',1,CONVERT(varchar(10), getdate(), 120 ),NULL),  (1,'K640','武汉','襄樊',321,'09:40',5,CONVERT(varchar(10), dateadd(day,7,getdate()), 120 ),NULL),  (1,'K140','武汉','常德',430,'18:50',4,CONVERT(varchar(10), dateadd(day,7,getdate()), 120 ),NULL),  (5,'Z841','武汉','景德镇',380,'13:40',4,CONVERT(varchar(10), getdate(), 120 ),NULL)  --增加一个今天出发 车型为现代  insert into routeinfo values(5,'Z841','武汉','景德镇',380,'19:40',4,CONVERT(varchar(10), getdate(), 120 ),NULL)  insert into Ticket  values(1,40,40,140),(2,40,40,60),(3,20,20,120),  (6,20,20,70),(7,20,20,80)  数据截图：  实验数据 | | | | | |
| 实验总结 | 此次实验中，完成一个小小的车站数据库模型。   1. 加深了E-R图的创建，关系模型的设计，通过E-R图和关系模型的设计了解了数据库中的关系，1对多，多对1，熟悉从E-R图到关系模型图再到数据表建立表模式。 2. 加深了视图对数据库安全性的重要作用，让不是很重要的用户看到了一些可以看的数据如：乘客只能查询发车信息。 3. 加深了使用连接查询、子查询、数据库中使用日期与字符串类型转换、灵活获取想要的日期和获取某一天的日期。 | | | | | |

|  |  |
| --- | --- |
| 指导教师意  见 | 签名： 年 月 日 |