**TUESDAY JANUARY 31**

use master;

drop database if exists SqlTutorial;

create database SqlTutorial;

use SqlTutorial;

create table [Customer] (

Id int not null primary key identity(1,1),

Name nvarchar(50) not null,

City nvarchar(50) not null,

State nvarchar(2) not null,

IsCorpAcct bit not null default 0,

CreditLimit int not null default 0,

Active bit not null default 1

);

Insert into [Customer] (Name, City, State, IsCorpAcct, CreditLimit)

values ('Mercury LTD', 'Newport', 'KY', 0, 100000);

Insert into [Customer] (Name, City, State, IsCorpAcct, CreditLimit)

values ('Venus LLC', 'Dayton', 'OH', 0, 300000);

Insert into [Customer] (Name, City, State, IsCorpAcct, CreditLimit)

values ('Mars INC', 'Cincinnati', 'OH', 1, 1000000);

Insert into [Customer] (Name, City, State, IsCorpAcct, CreditLimit)

values ('Jupiter LPA', 'Covington', 'KY', 0, 250000);

Insert into [Customer] (Name, City, State, IsCorpAcct, CreditLimit)

values ('Saturn INC', 'Batesville', 'IN', 0, 300000);

Insert into [Customer] (Name, City, State, IsCorpAcct, CreditLimit)

values ('Uranus LPA', 'Indianapolis', 'IN', 1, 1000000);

Insert into [Customer] (Name, City, State, IsCorpAcct, CreditLimit)

values ('Neptune INC', 'Cleveland', 'OH', 1, 1000000);

create table [Order] (

Id int not null primary key identity(1,1),

Date datetime not null,

Amount decimal(14,0) not null default 0,

CustomerId int foreign key references Customer

);

Insert into [Order] (Date, Amount, CustomerId)

values ('2017-12-31', 778, 6);

Insert into [Order] (Date, Amount, CustomerId)

values ('2017-7-3', 156, 5);

Insert into [Order] (Date, Amount, CustomerId)

values ('2017-6-23', 491, 7);

Insert into [Order] (Date, Amount, CustomerId)

values ('2017-3-6', 230, 6);

Insert into [Order] (Date, Amount, CustomerId)

values ('2017-11-21', 844, 5);

Insert into [Order] (Date, Amount, CustomerId)

values ('2017-9-18', 849, 2);

Insert into [Order] (Date, Amount, CustomerId)

values ('2017-4-15', 371, 2);

Insert into [Order] (Date, Amount, CustomerId)

values ('2017-7-13', 451, 2);

Insert into [Order] (Date, Amount, CustomerId)

values ('2017-1-7', 510, 4);

Insert into [Order] (Date, Amount, CustomerId)

values ('2017-9-23', 612, 6);

Insert into [Order] (Date, Amount, CustomerId)

values ('2017-10-11', 736, 5);

select c.id, name as 'Customer Name', date, amount

from customer c

left join [order] o

on o.customerid = c.id

order by o.amount desc

-- display all orders with amounts < max order amount in oh

select customerid, date, amount from [order]

where amount < (select max(amount) from [order] o

join customer c

on o.CustomerId = c.id

where c.state = 'OH')

order by amount desc

**WEDNESDAY JANUARY 31**

select concat(s.firstname, ' ', s.lastname) as 'Name', c.description as 'Class'

from student s

join schedule sc

on s.id = sc.StudentId

join class c

on c.id = sc.classid

where s.firstname = 'Aaron' and s.LastName = 'Zell'

OUTPUT:

Name Class

Aaron Zell English 103

Aaron Zell Math 102

Aaron Zell American History

Aaron Zell Information Systems 101

--Create a list of all students and their majors.

--Display the FirstName, LastName, SAT, GPA from the student and Description from Major using an alias of Major

--Example:

--FirstName LastName SAT GPA Major

--Greg Doud 1600 3.5 Information Systems

select s.FirstName, s.LastName, s.SAT, s.GPA, m.Description

from student s

join major m

on s.MajorId = m.id

order by S.GPA desc

OUTPUT:

FirstName LastName SAT GPA Description

Bill Yatil 1420 3.7 Management

Aaron Zell 1250 3.2 Math

Cory Xe 1120 3.1 Business

Devin Waite 1030 2.9 Communications

--display FirstName, LastName from student taking class Math 102

--Example:

--FirstName LastName

--Greg Doud

select s.FirstName, s.LastName from class c

join schedule sc on sc.ClassId = c.id

join student s on s.id = sc.StudentId

where c.Description = 'Math 102'

OUTPUT

Aaron Zell

--display all classes for students with a major of Math

--Example:

--FirstName LastName Class

--Greg Doud Math

select concat(s.firstname, ' ', s.lastname) as 'Name', c.description as 'Class'

from student s

join major m on m.id = s.majorid

join schedule sc on sc.studentid = s.id

join class c on c.id = sc.classid

where m.description = 'Math'

update class set

Description = Description + ' 301'

where id in (7, 8)

ALSO

Where id in (select id from class where INSTR(‘Economics’ in description) > 0)

select \* from class

OUTPUT:

Name Class

Aaron Zell English 103

Aaron Zell Math 102

Aaron Zell American History

Aaron Zell Information Systems 101

OUTPUT:

Id Description

1 English 101

2 English 102

3 English 103

4 Math 101

5 Math 102

6 Math 103

7 Micro Economics 301

8 Macro Economics 301

9 American History

10 World History

11 European History

12 Information Systems 101

13 Information Systems 102

14 Information Systems 103

**THURSDAY FEBRUARY 1**

CREATE TABLE ToDo (

id int primary key identity (1,1),

Task nvarchar(80) not null,

Priority int not null default 5

check (Priority > 0 and Priority < 10),

Completed bit not null default 0,

DueDate datetime not null,

CategoryId int foreign key references Category(Id)

)

CREATE TABLE Category (

Id int primary key identity (1,1),

Description varchar(25) not null

)

INSERT into Todo (task, duedate, categoryid)

Values (‘Review Lisa’’s project’, ‘2018-2-1 12:00:00’, 2)

**EXERCISE**

use master

go

drop database if exists Corporate

go

create database Corporate

go

use Corporate

go

drop table if exists Employee

drop table if exists Manager

drop table if exists Department

create table Department(

Id int primary key identity(1,1),

DeptName nvarchar(50) not null,

CostCenter int not null

check(CostCenter >= 100000 and CostCenter <= 999999 ),

Active bit not null default 1,

)

insert into department (DeptName, CostCenter) values ('AR Dept', 300000)

create table Manager(

Id int primary key identity(1,1),

FirstName nvarchar(50) not null,

LastName nvarchar(50) not null,

DepartmentId int not null foreign key references Department(Id)

)

insert into Manager (FirstName, LastName, DepartmentId) values ('Greg', 'Doud', 1)

create table Employee(

Id int not null primary key identity(1,1),

FirstName nvarchar(50) not null,

LastName nvarchar(50) not null,

Birthday datetime NOT NULL,

Job nvarchar(50) not null,

ManagerId int not null foreign key references Manager(Id)

)

insert into Employee (FirstName, LastName, Birthday, Job, ManagerId)

values ('Michael', 'Robinson', '08-04-1965', 'Sr Prog Analyst', 1)

select \* from Department

select \* from Manager

select \* from Employee

select CONCAT(e.FirstName, ' ', e.LastName) as 'Employee',

e.Birthday,

e.Job,

CONCAT(m.FirstName, ' ', m.LastName) as 'Manager',

d.DeptName as 'Department',

d.CostCenter

from Employee e

join Manager m

on m.id = e.ManagerId

join Department d

on d.Id = m.DepartmentId