

**Database Management Systems**  
**COSC 4385.001**  
**Spring 2020**  
**Dr. Leonard Brown**  
**Timothy Norrell**

**Virtual World Database**

## Table of Contents

[illegible]

## E-R Diagram



## Database Dictionary

Territory: tID, ClimateDescription, tName

```
{
tID: Numeric/ The ID of the Territory/ Primary Key, must be positive, not null, unique
ClimateDescription: String/ Description of the climate in the territory/ not null
tName: String/ The name of the territory/ not null
}
```

Settlement: sName, tID, XCoordinate, YCoordinate, ElevationLevel, DateFounded

```
{
sName: String/Name of the settlement/ Primary Key, not null, unique
tID: Numeric/ The ID of the Territory/ Foreign Key, must be positive, not null, unique
XCoordinate: Numeric/ The X Coordinate of the Location/ not null
YCoordinate: Numeric/ The Y Coordinate of the Location/ not null
ElevationLevel: Numeric/ The Elevation Level Coordinate of the Location/ not null
DateFounded: String/ The date the settlement was founded/ not null
}
```

City: sName, Mayor, Population

```
{
sName: String/Name of the settlement/ Foreign Key, not null, unique
Mayor: String/ Name of the Mayor/ not null
Population: Numeric/ Number of people living at the settlement/ not null, must be positive
}
```

Village: sName, RiverName

```
{
sName: String/Name of the settlement/ Foreign Key, not null, unique
RiverName: String/Name of the river the settlement is next to/ not null
}
```

TheUser: Email, Username, Motto, Logo

```
{
Email: String/The email of the user/ Primary Key, not null, unique
Username: String/ The username of the user/ not null
Motto: String/ The motto of the user/ not null
Logo: String/ Description of the logo the user uses/ not null
}
```

TheCharacter: CharacterName, Email, PointTotal, Species

```
{
CharacterName: String/ The name of the character/ Primary Key, not null, unique
Email: String/ The email that character belongs to/ Foreign Key, not null, unique
PointTotal: Numeric/ The number of experience points the character has/ must be positive, not
null
Species: String/ The species the character is/ not null
}
```

Skills: CharacterName, Skills

```
{
CharacterName: String/ The name of the character these skills belong to/ Foreign Key, not null,
unique
Skills: String/ The skill the character has/ Primary Key, not null
}
```

Country: cName, Government, sName, Leader

```
{
cName: String/ The name of the country/ Primary Key, not null, unique
Government: String/ The form of government in the country/ not null
sName: String/ The name of the capital for the country/ Foreign Key, not null, unique
Leader String/ The name of the leader in the country/ not null
}
```

Resources: cName, Resources, Lending

```
{
cName: String/ The name of the country these resources are in/ Foreign Key, not null, unique
Resources: String/ The name of the resources/ Primary Key, not null
Lending: String/ Status of whether this resource is being lendd/ not null
}
```

CharacterResidence: CharacterName, sName, tID, ResidencyDate, Address

```
{
CharacterName: String/ The name of the character living here/ Foreign Key, not null, unique
sName: String/ The name of the settlement the character is living in/Foreign Key, not null,
unique
tID: Numeric/ The territory ID the settlement is in/ Foreign Key, not null, unique, must be
positive
ResidencyDate: String/ The date the character moved in/ not null
Address: String/ The address of the living space/ not null
}
```

TerritoryResidence: tID, cName

{

tID: Numeric/ The territory ID that is in the country/ Foreign Key, not null, must be positive, unique

cName: String/ The name of the country that has this territory ID/ Foreign Key, not null, unique

}

## SQL Statements for Queries

### Query 1:

```
create table Territory (  
  tID number (10),  
  ClimateDescription varchar2 (20),  
  tName varchar2 (20),  
  primary key (tID),  
  check (tID between 0 and 9999999999)  
);
```

```
create table Settlement(  
  sName varchar2 (20),  
  tID number (10),  
  XCoordinate number,  
  YCoordinate number,  
  ElevationLevel number,  
  DateFounded varchar2 (10),  
  primary key (sName),  
  foreign key (tID) references Territory on delete cascade  
);
```

```
create table City (  
  sName varchar2 (20),  
  Mayor varchar2 (20),  
  Population varchar2 (20),  
  foreign key (sName) references Settlement on delete cascade  
);
```

```
create table Village (  
  sName varchar2 (20),  
  RiverName varchar2 (20),  
  foreign key (sName) references Settlement on delete cascade  
);
```

```
create table TheUser(  
  Email varchar2 (20),  
  Username varchar2 (20),  
  Motto varchar2 (20),  
  Logo varchar2 (20),  
  primary key (Email)
```

```
);
```

```
create table TheCharacter (
  CharacterName varchar2 (20),
  Email varchar2 (20),
  PointTotal number,
  Species varchar2 (20),
  primary key (CharacterName),
  foreign key (Email) references TheUser on delete cascade,
  check (PointTotal >= 0)
);
```

```
create table Skills (
  CharacterName varchar2 (20),
  Skills varchar2 (20),
  primary key (Skills),
  foreign key (CharacterName) references TheCharacter on delete cascade
);
```

```
create table Country (
  cName varchar2 (20),
  Government varchar2 (20),
  sName varchar2 (20),
  Leader varchar2 (20),
  primary key (cName),
  Foreign key (sName) references Settlement on delete cascade
);
```

```
create table Resources (
  cName varchar2 (20),
  Resources varchar2 (20),
  Lending varchar2 (3),
  primary key (Resources),
  foreign key (cName) references Country on delete cascade
);
```

```
create table CharacterResidence (
  CharacterName varchar2(20),
  sName varchar (20),
  tID varchar (10),
  ResidencyDate varchar (10),
  Address varchar (20),
  foreign key (CharacterName) references TheCharacter on delete cascade,
```



```
foreign key (sName) references Settlement on delete cascade,  
foreign key (tID) references Settlement on delete cascade  
);
```

```
create table TerritoryResidence (  
tID number (10),  
cName varchar2 (20),  
foreign key (tID) references Territory on delete cascade,  
foreign key (cName) references Country on delete cascade  
);
```

**Query 2:**

```
insert into Territory (tID, ClimateDescription, tName)
values ('1836927372', 'Super chilly', 'ChillyLand');
```

```
insert into Territory (tID, ClimateDescription, tName)
values ('0123841098', 'Super sunny', 'SunnyLand');
```

```
insert into Territory (tID, ClimateDescription, tName)
values ('1849205638', 'Super rainy', 'RainyLand');
```

```
insert into Territory (tID, ClimateDescription, tName)
values ('2849005638', 'Super foggy', 'FoggyLand');
```

```
insert into Settlement (sName, tID, XCoordinate, YCoordinate, ElevationLevel, DateFounded)
values ('Boingo', '1836927372', '354.1892', '198.0029', '87.1332', '10/21/1878');
```

```
insert into Settlement (sName, tID, XCoordinate, YCoordinate, ElevationLevel, DateFounded)
values ('Toronbo', '1836927372', '618.2919', '129.2983', '101.1972', '6/1/1593');
```

```
insert into Settlement (sName, tID, XCoordinate, YCoordinate, ElevationLevel, DateFounded)
values ('Hamlet', '0123841098', '139.0183', '687.1056', '97.1023', '2/7/1672');
```

```
insert into Settlement (sName, tID, XCoordinate, YCoordinate, ElevationLevel, DateFounded)
values ('Gogone', '0123841098', '719.1834', '172.1832', '123.9283', '7/18/1846');
```

```
insert into Settlement (sName, tID, XCoordinate, YCoordinate, ElevationLevel, DateFounded)
values ('Pinkton', '1849205638', '278.1029', '592.1281', '99.1038', '1/2/1717');
```

```
insert into Settlement (sName, tID, XCoordinate, YCoordinate, ElevationLevel, DateFounded)
values ('Yak', '1849205638', '744.2718', '654.1999', '66.7888', '7/29.1923');
```

```
insert into Settlement (sName, tID, XCoordinate, YCoordinate, ElevationLevel, DateFounded)
values ('Opin', '1849205638', '443.6729', '111.0032', '77.3939', '3/25/1700');
```

```
insert into City (sName, Mayor, Population)
values ('Boingo', 'Ted Bungo', '2374');
```

```
insert into City (sName, Mayor, Population)
values ('Toronbo', 'Shila Namm', '474');
```

```
insert into City (sName, Mayor, Population)
values ('Hamlet', 'Green Ham', '1001');
```

```
insert into Village (sName, RiverName)
values ('Gogone', 'Viven');
```

```
insert into Village (sName, RiverName)
values ('Pinkton', 'Trish');
```

```
insert into Village (sName, RiverName)
values ('Yak', 'Brash');
```

```
insert into TheUser (Email, Username, Motto, Logo)
values ('JojoFan1@gmail.com', 'JojoFan1', 'Yare Yare Daze', 'A hat');
```

```
insert into TheUser (Email, Username, Motto, Logo)
values ('JojoFan2@gmail.com', 'JojoFan2', 'ZaWarudo', 'Pose');
```

```
insert into TheUser (Email, Username, Motto, Logo)
values ('JojoFan3@gmail.com', 'JojoFan3', 'Yare Yare Daze', 'Dog Face');
```

```
insert into TheCharacter (CharacterName, Email, PointTotal, Species)
values ('Jojo', 'JojoFan1@gmail.com', '78431', 'Human');
```

```
insert into TheCharacter (CharacterName, Email, PointTotal, Species)
values ('Dio', 'JojoFan2@gmail.com', '55182', 'Human');
```

```
insert into TheCharacter (CharacterName, Email, PointTotal, Species)
values ('Iggy', 'JojoFan3@gmail.com', '99999', 'Dog');
```

```
insert into Skills (CharacterName, Skills)
values ('Jojo', 'Charisma');
```

```
insert into Skills (CharacterName, Skills)
values ('Jojo', 'Science');
```

```
insert into Skills (CharacterName, Skills)
values ('Dio', 'Time-Stopping');
```

```
insert into Skills (CharacterName, Skills)
values ('Iggy', 'Barking');
```

```
insert into Country (cName, Government, sName, Leader)
```

```
values ('Haendall', 'Democracy', 'Boingo', 'Alias Fendrear');
```

```
insert into Country (cName, Government, sName, Leader)
values ('Falon', 'Monarchy', 'Hamlet', 'King Jergan');
```

```
insert into Country (cName, Government, sName, Leader)
values ('Chillrend', 'Oligarchy', 'Toronbo', 'Hammond Rock');
```

```
insert into Resources (cName, Resources, Lending)
values ('Haendall', 'Cotton', 'Yes');
```

```
insert into Resources (cName, Resources, Lending)
values ('Haendall', 'Sugar', 'Yes');
```

```
insert into Resources (cName, Resources, Lending)
values ('Falon', 'Turnips', 'No');
```

```
insert into Resources (cName, Resources, Lending)
values ('Chillrend', 'Bananas', 'Yes');
```

```
insert into CharacterResidence (CharacterName, sName, tID, ResidencyDate, Address)
values ('Jojo', 'Boingo', '', '6/3/1732', '1234 Bingo Road');
```

```
insert into CharacterResidence (CharacterName, sName, tID, ResidencyDate, Address)
values ('Jojo', 'Toronbo', '', '6/5/1732', '1534 Dingo Road');
```

```
insert into CharacterResidence (CharacterName, sName, tID, ResidencyDate, Address)
values ('Iggy', 'Hamlet', '', '6/3/1732', '555 Rally Road');
```

```
insert into TerritoryResidence (tID, cName)
values ('1836927372', 'Haendall');
```

```
insert into TerritoryResidence (tID, cName)
values ('0123841098', 'Haendall');
```

```
insert into TerritoryResidence (tID, cName)
values ('1849205638', 'Falon');
```

```
insert into TerritoryResidence (tID, cName)
values ('2849005638', 'Chillrend');
```

**Query 3:**

```

select y.cName, y.Leader
from
  Country y,
  (select cName, count(*) numm
   from Resources
   where Lending = 'Yes'
   group by cName) x,
  (select max(numm) most
   from (select cName, count(*) numm
        from Resources
        where Lending = 'Yes'
        group by cName)
   ) b
where x.numm = b.most
and y.cName = x.cName;

```

**Query 4:**

```

select xx.sName, s.ElevationLevel, ter.ClimateDescription
from Country q,
Territory ter,
Settlement s,
(select x.sName
 from City x,
  (select max(Population) mp from City y) b
 where x.Population = b.mp) xx
where xx.sName = q.sName
and s.sName = xx.sName
and ter.tID = s.tID

```

**Query 5:**

```

select tc.CharacterName, tc.PointTotal/2000 TheLevel
from TheCharacter tc,
    (select CharacterName, count(CharacterName)
     from CharacterResidence
     group by CharacterName
     having count(CharacterName) > 1) cs
where tc.CharacterName = cs.CharacterName

```

**Query 6:**

```

select cy.cName Country, cy.Leader, cy.sName Capital, cn.x NumberofTerritories
from Country cy,
    (select cName, count(cName) x
     from TerritoryResidence
     group by cName) cn
where cn.cName = cy.cName;

```

**Query 7:**

```

select Username, CharacterName, Species
from TheCharacter ch,
    TheUser u,
    (select max(PointTotal) mp from TheCharacter) x
where u.email = ch.email
and PointTotal = x.mp;

```

**Query 8:**

```
update TheCharacter
set PointTotal = PointTotal * 2
where CharacterName in (select cs.CharacterName
                        from
                          (select CharacterName, Skills
                           from Skills s
                           where s.Skills = 'Charisma'
                           or s.Skills = 'Science') cs
                        group by CharacterName
                        having count(Skills) = 2)
```

**Query 9:**

```
delete from Settlement
where sName not in (select sName from CharacterResidence)
```

**Query 10:**

```
drop table TerritoryResidence;
drop table CharacterResidence;
drop table Resources;
drop table Country;
drop table Skills;
drop table TheCharacter;
drop table TheUser;
drop table Village;
drop table City;
drop table Settlement;
drop table Territory;
```