

### **Insert Army**

```
AuthorityModel commanderInChief = new AuthorityModel("King Yoloman");  
insertAuthority(commanderInChief);
```

```
ArmyModel armyInsert = new ArmyModel("Golden Armada", "King Yoloman");  
insertArmy(arrayInsert );
```

### **Insert Authority**

```
AuthorityModel insertAuthority = new AuthorityModel("King Pastaman");  
insertAuthority(insertAuthority);
```

### **Insert Holiday**

```
HolidayModel xmas = new Holiday("Christmas", "Catholicism", 25-DEC-2021);  
insertHoliday(xmas);
```

### **Insert Citizen**

```
CitizenModel insertCitizen1 = new CitizenModel(1, null, "Catholicism", "Payton", 1);  
insertCitizen(insertCitizen1 );
```

### **Insert Celebrates**

```
CitizenModel citizenCelebrates1 = new CitizenModel(100, null, "Pastafarian", "Taylor", 1);  
insertCitizen(citizenCelebrates1 );
```

```
HolidayModel pastaDayCelebrates = new Holiday("Pasta Day", "Pastafarian", 21-APR-2021);  
insertHoliday(pastaDayCelebrates );
```

```
CelebratesModel celebrate1 = new CelebratesModel (100, "Pastafarian", "PastaDay");  
insertCelebrates(celebrate1);
```

### **Insert TributePayments**

```
CitizenModel tributePayingCitizen1 = new CitizenModel(51, null, "Catholicism", "Dayton", 1);  
insertCitizen(tributePayingCitizen1)
```

```
AuthorityModel tributeReceivingCitizen1 = new AuthorityModel("King Darthur");  
insertAuthority(tributeReceivingCitizen1);
```

```
TributePaymentsModel tributePaymentInsert1 = new TributePaymentsModel(51, "King Darthur",  
"Horse", 2);
```

```
insertTributePayment(tributePaymentInsert1);
```

### **Delete Citizen**

```
CitizenModel citizenToDelete1 = new CitizenModel(2000, null, "Catholicism", "Layton", 1);  
CitizenModel citizenToDelete2 = new CitizenModel(2001, null, "Catholicism", "Zayton", 1);
```

```
insertCitizen(citizenToDelete1 )  
insertCitizen(citizenToDelete2 )
```

```
deleteCitizen(citizenToDelete1);
```

Check to see if citizenToDelete1 is gone but citizenToDelete2 is still present

### **Update Citizen**

NOTE THAT THIS TEST MUST RUN THROUGH THE UI; It doesn't make any sense as a straight up unit test

```
AuthorityModel authorityProject = new AuthorityModel("Prince Arthas");  
ArmyModel armyProject = new ArmyModel("The Scourge")
```

```
CitizenModel citizenUpdate = new CitizenModel(1300, "The Scourge", null, "Abomination", 2);
```

```
insertAuthority(authorityProject);  
insertArmy(armyProject);  
insertCitizen(citizenUpdate);
```

```
CitizenModel citizenUpdateNew1 = new CitizenModel(1300, "The Scourge", null, "Abomination",  
2);  
CitizenModel citizenUpdateNew2 = new CitizenModel(1300, "The Forsaken", null,  
"Abomination", 2);  
CitizenModel citizenUpdateNew3 = new CitizenModel(1300, "The Forsaken", null,  
"Abomination", 3);  
CitizenModel citizenUpdateNew4 = new CitizenModel(1300, "The Forsaken", null, "Plague  
Doctor", 3);
```

JUST CHECK TO SEE IF THEY UPDATE

### **Project Citizen**

```
AuthorityModel authorityProject = new AuthorityModel("Griffith");
ArmyModel armyProject = new ArmyModel("Band of the Hawk");
```

```
CitizenModel citizenProject1 = new CitizenModel(1200, "Band of the Hawk", null, "Guts", 9);
CitizenModel citizenProject2 = new CitizenModel(1201, "Band of the Hawk", null, "Casca", 8);
CitizenModel citizenProject3 = new CitizenModel(1202, "Band of the Hawk", null, "Judeau", 7);
CitizenModel citizenProject4 = new CitizenModel(1203, "Band of the Hawk", null, "Pippin", 7);
CitizenModel citizenProject5 = new CitizenModel(1204, "Band of the Hawk", null, "Rickert", 7);
CitizenModel citizenProject6 = new CitizenModel(1205, "Band of the Hawk", null, "Corkus", 7);
```

```
insertAuthority(authorityProject);
insertArmy(armyProject);
insertCitizen(citizenProject1);
insertCitizen(citizenProject2);
insertCitizen(citizenProject3);
insertCitizen(citizenProject4);
insertCitizen(citizenProject5);
insertCitizen(citizenProject6);
```

Just choose various projections to see if they work

### **Citizen Join Holiday**

```
AuthorityModel requiredAuthorityJoin = new Authority("Steve");
insertAuthority(requiredAuthorityJoin);
```

```
ArmyModel requiredArmyJoin = new ArmyModel("Golden Armada", "Steve");;
insertArmy(requiredArmyJoin);
```

```
CitizenModel citizenJoin1 = new CitizenModel(800, "Golden Armada", "Catholicism", "Peter", 1);
CitizenModel citizenJoin2 = new CitizenModel(801, "Golden Armada", "Catholicism", "Peter", 1);
CitizenModel citizenJoin3 = new CitizenModel(802, "Golden Armada", "Paganism", "Peter", 1);
CitizenModel citizenJoin4 = new CitizenModel(803, "Golden Armada", "Shamanism", "Peter", 1);
```

```
insertCitizen (citizenJoin1);
insertCitizen (citizenJoin2);
insertCitizen (citizenJoin3);
insertCitizen (citizenJoin4);
```

```
HolidayModel holidayJoin1 = new Holiday("Christmas", "Catholicism", 12-DEC-2021);
HolidayModel holidayJoin2 = new Holiday("Easter", "Catholicism", 04-APR-2021);
HolidayModel holidayJoin3 = new Holiday("Soul Harvesting", "Shamanism", 02-JAN-2021);
```

```
HolidayModel holidayJoin4 = new Holiday("Blood sacrifices", "Paganism", 03-JAN-2021);
```

```
insertHoliday(holidayJoin1);  
insertHoliday(holidayJoin2);  
insertHoliday(holidayJoin3);  
insertHoliday(holidayJoin4);
```

```
CelebratesModel celebratesJoin1 = new CelebratesModel(800, "Catholicism", "Christmas");  
CelebratesModel celebratesJoin2 = new CelebratesModel(800, "Catholicism", "Easter");  
CelebratesModel celebratesJoin3 = new CelebratesModel(801, "Catholicism", "Christmas");  
CelebratesModel celebratesJoin4 = new CelebratesModel(802, "Paganism", "Blood sacrifices");  
CelebratesModel celebratesJoin5 = new CelebratesModel(803, "Shamanism", "Soul  
Harvesting");
```

```
insertCelebrates(celebratesJoin1);  
insertCelebrates(celebratesJoin2);  
insertCelebrates(celebratesJoin3);  
insertCelebrates(celebratesJoin4);  
insertCelebrates(celebratesJoin5);
```

```
Case "h.AssociatedReligion = Catholicism" -> 800, 801
```

```
Case "h.Name = Christmas" -> 800, 801
```

```
Case "h.Name = Easter" -> 800
```

```
Case "h.Date = 03-JAN-2021" -> 802
```

### **Aggregation Group By**

```
CitizenModel citizenHaving1 = new CitizenModel(300, null, "Catholicism", "Pam", 1);  
CitizenModel citizenHaving2 = new CitizenModel(301, null, "Catholicism", "Sam", 1);  
CitizenModel citizenHaving3 = new CitizenModel(302, null, "Catholicism", "Cam", 1);
```

```
CitizenModel citizenHaving4 = new CitizenModel(303, null, "Pastafarian", "Derek", 1);  
CitizenModel citizenHaving5 = new CitizenModel(304, null, "Pastafarian", "Derrick", 1);
```

```
CitizenModel citizenHaving6 = new CitizenModel(305, null, "Protestant", "Paul", 1);
```

```
insertCitizen(citizenhaving1);  
insertCitizen(citizenhaving2);  
insertCitizen(citizenhaving3);  
insertCitizen(citizenhaving4);  
insertCitizen(citizenhaving5);  
insertCitizen(citizenhaving6);
```

```
Expect {
```

```
("Catholicism", 3),  
("Pastafarian", 2),  
("Protestant", 1)  
}
```

## Aggregation Having

```
AuthorityModel theKing = new Authority("King DatKing");  
AuthorityModel thePrince = new Authority("Prince DatPrince");
```

```
CitizenModel citizenHaving1 = new CitizenModel(700, null, "Catholicism", "Peter", 1);  
CitizenModel citizenHaving2 = new CitizenModel(701, null, "Catholicism", "Oscar", 1);
```

```
TributePaymentsModel tributeToTheKing1 = new TributeModel(700, "King DatKing", "Hens", 5);  
TributePaymentsModel tributeToTheKing2 = new TributeModel(701, "King DatKing", "Pigs", 9);  
TributePaymentsModel tributeToThePrince = new TributeModel(700, "Prince DatPrince", "Gold  
Coins", 2);
```

```
Given n = 1, Expect { ("King DatKing", 2) , ("Prince DatPrince", 1) }  
Given n = 2, Expect { ("King DatKing", 2)}
```

## Nested Aggregation

```
AuthorityModel authorityNested = new AuthorityModel("Aragorn");  
insertAuthority(authorityNested);
```

```
ArmyModel requiredArmyNested = new ArmyModel("Golden Armada", "Aragorn");  
insertArmy(requiredArmyNested);
```

```
CitizenModel citizenDivision1 = new CitizenModel(900, "Golden Armada", "Catholicism", "Peter",  
1);  
CitizenModel citizenDivision2 = new CitizenModel(901, "Golden Armada", "Catholicism", "Peter",  
1);
```

```
TributePaymentsModel tributePaymentsNested1 = new TributePaymentsModel(900, "Aragorn",  
"Hens", 5);  
TributePaymentsModel tributePaymentsNested2 = new TributePaymentsModel(901, "Aragorn",  
"Hens", 15);
```

```
CitizenModel citizenDivision3 = new CitizenModel(902, "Golden Armada", "Catholicism", "Peter",  
2);  
CitizenModel citizenDivision4 = new CitizenModel(903, "Golden Armada", "Catholicism", "Peter",  
2);
```

```
TributePaymentsModel tributePaymentsNested3 = new TributePaymentsModel(902, "Aragorn",  
"Hens", 15);
```

```
TributePaymentsModel tributePaymentsNested4 = new TributePaymentsModel(903, "Aragorn",  
"Hens", 25);
```

```
CitizenModel citizenDivision5 = new CitizenModel(904, "Golden Armada", "Catholicism", "Peter",  
3);
```

```
CitizenModel citizenDivision6 = new CitizenModel(905, "Golden Armada", "Catholicism", "Peter",  
3);
```

```
TributePaymentsModel tributePaymentsNested5 = new TributePaymentsModel(904, "Aragorn",  
"Hens", 25);
```

```
TributePaymentsModel tributePaymentsNested6 = new TributePaymentsModel(905, "Aragorn",  
"Hens", 35);
```

```
insertCitizen(citizenDivision1);
```

```
insertCitizen(citizenDivision2);
```

```
insertCitizen(citizenDivision3);
```

```
insertCitizen(citizenDivision4);
```

```
insertCitizen(citizenDivision5);
```

```
insertCitizen(citizenDivision6);
```

```
insertPaysTribute(tributePaymentsNested1);
```

```
insertPaysTribute(tributePaymentsNested2);
```

```
insertPaysTribute(tributePaymentsNested3);
```

```
insertPaysTribute(tributePaymentsNested4);
```

```
insertPaysTribute(tributePaymentsNested5);
```

```
insertPaysTribute(tributePaymentsNested6);
```

Expect

(1, 10)

(2, 20)

(3, 30)

## **Division**

```
AuthorityModel authorityDivision1 = new AuthorityModel("King Dominik");
```

```
insertAuthority(authorityDivision1);
```

```
AuthorityModel authorityDivision2 = new AuthorityModel("Napoleon");
```

```
insertAuthority(authorityDivision2);
```

```
AuthorityModel authorityDivision3 = new AuthorityModel("Reinhard von Lohengramm");
```

```
insertAuthority(authorityDivision3);
```

```
ArmyModel requiredArmyDivision = new ArmyModel("Golden Armada", "King Dominik");
```

```
insertArmy(requiredArmyDivision);
```

```
CitizenModel citizenDivision1 = new CitizenModel(1000, "Golden Armada", "Catholicism",  
"Peter", 1);
```

```
insertCitizen(citizenDivision1);
```

```
CitizenModel citizenDivision2 = new CitizenModel(1001, "Golden Armada", "Catholicism",  
"Peter", 1);
```

```
insertCitizen(citizenDivision2);
```

```
CitizenModel citizenDivision3 = new CitizenModel(1002, "Golden Armada", "Catholicism",  
"Peter", 1);
```

```
insertCitizen(citizenDivision3);
```

```
CitizenModel citizenDivision4 = new CitizenModel(1003, "Golden Armada", "Catholicism",  
"Peter", 1);
```

```
insertCitizen(citizenDivision4);
```

```
TributePaymentsModel tributePaymentsDivision1 = new TributePaymentsModel(1000, "King  
Dominik", "Hens", 20);
```

```
TributePaymentsModel tributePaymentsDivision2 = new TributePaymentsModel(1000,  
"Napoleon", "Hens", 20);
```

```
TributePaymentsModel tributePaymentsDivision3 = new TributePaymentsModel(1000,  
"Reinhard von Lohengramm", "Hens", 20);
```

```
TributePaymentsModel tributePaymentsDivision4 = new TributePaymentsModel(1001, "King  
Dominik", "Hens", 20);
```

```
TributePaymentsModel tributePaymentsDivision5 = new TributePaymentsModel(1001,  
"Napoleon", "Hens", 20);
```

```
TributePaymentsModel tributePaymentsDivision6 = new TributePaymentsModel(1002,  
"Napoleon", "Hens", 20);
```

```
TributePaymentsModel tributePaymentsDivision7 = new TributePaymentsModel(1002,  
"Reinhard von Lohengramm", "Hens", 20);
```

```
TributePaymentsModel tributePaymentsDivision8 = new TributePaymentsModel(1003, "King  
Dominik", "Hens", 20);
```

```
TributePaymentsModel tributePaymentsDivision9 = new TributePaymentsModel(1003,  
"Reinhard von Lohengramm", "Hens", 20);
```

```
insertPaysTribute(tributePaymentsDivision1);
```

```
insertPaysTribute(tributePaymentsDivision2);
```

```
insertPaysTribute(tributePaymentsDivision3);
```

```
insertPaysTribute(tributePaymentsDivision4);
```

```
insertPaysTribute(tributePaymentsDivision5);
```

```
insertPaysTribute(tributePaymentsDivision6);
```

```
insertPaysTribute(tributePaymentsDivision7);
```

```
insertPaysTribute(tributePaymentsDivision8);  
insertPaysTribute(tributePaymentsDivision9);
```

Expect ONLY 1000 to display