```
assert_eq_relname  THM

⊢  ∀a,b:relname(). eq_relname(a;b) ⟺ a = b
| by UnivCD THENA Auto THEN Unfolds ``relname eq_relname`` 0

|    THEN D 2 THEN D 1 THEN Reduce 0 THEN
Try (RWW "assert_st_eq" 0)
|
THEN Auto
THEN EqLblFwd
THEN Try (EqCD THEN Complete Auto)
|
|\
| 1. x1 : SimpleType
| 2. x : SimpleType
| 3. inl(x1) = inl(x)
| ⊢  x1 = x
|   by ApFunToHypEquands `z' Case(z)
|                             Case eq(a) =>
|                                 a
|                             Default => x SimpleType (-1)
|
|      THENM Reduce (-1)
| THEN Auto
| THEN D (-1) THEN Reduce 0
|
|      THEN Auto
|
|\
| 1. y : Label
| 2. x : SimpleType
| 3. inr(y) = inl(x)
| ⊢  False
|   by ApFunToHypEquands `z' Case(z)
|                             Case eq(a) =>
|                                 0
|                             Default => 1 ℤ (-1)
| THENM Reduce (-1)
|
|      THEN Auto
| THEN D (-1) THEN Reduce 0
| THEN Auto
|
|\
| 1. x : SimpleType
| 2. y : Label
| 3. inl(x) = inr(y)
| ⊢  False
|   by ApFunToHypEquands `z' Case(z)
|                             Case eq(a) =>
|                                 0
|                             Default => 1 ℤ (-1)
| THENM Reduce (-1)
|
|      THEN Auto
```

```
| THEN D (-1) THEN Reduce 0
| THEN Auto
|
.\
. 1. y1 : Label
. 2. y : Label
. 3. inr(y1) = inr(y)
. ⊢  y1 =1 y
. | by ApFunToHypEquands ‘z’ Case(z)
. |                          Case a =>
. |                              a
. |                          Default => y Label (-1)
.
. |    THENM Reduce (-1)
. THEN Auto
. THEN D (-1) THEN Reduce 0
.
. |    THEN Auto
. |
. 1. y1 : Label
. 2. y : Label
. 3. inr(y1) = inr(y)
. 4. y1 = y
. ⊢  y1 =1 y  by HypSubst (-1) 0
. THENM EqLblReflexive 0
. THEN Auto
END
```