1. **Material(MaterialID, MaterialName, MaterialType)** ( E / WE / R / DA )

**District(DistrictID, DistrictName)** ( E / WE / R / DA )

**Unit(DistrictID, UnitID, UnitName)** ( E / WE / R / DA )

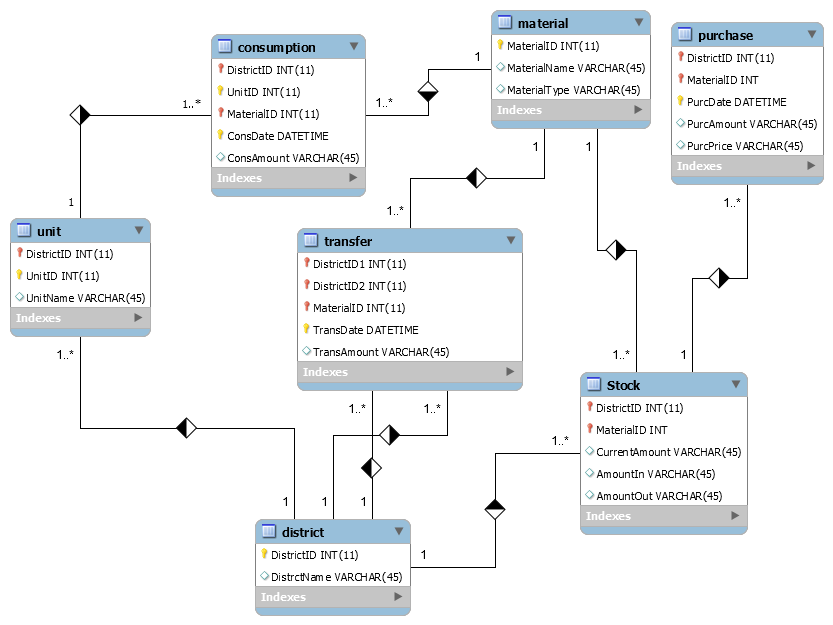
**Stock(DistrictID, MaterialID, CurrentAmount, AmountIn, AmountOut)** ( E / WE / R / DA )

**Purchase(DistrictID, MaterialID, PurcDate, PurcAmount, UnitPrice)** ( E / WE / R / DA )

**Consumption(DistrictID, UnitID, MaterialID, ConsDate, ConsAmount)** ( E / WE / R / DA )

**Transfer(DistrictID1, DistrictID2, MaterialID, TransDate, TransAmount)** ( E / WE / R / DA )

1. Right answers are marked with red color.



1. Find Id and name of the materials whose current amount is zero in at least two different districts.

X1←  DistrictID, MaterialID ((TotalAmount>0)(Stock))

X2←  MaterialID ((MaterialType= “food”)(Material))

X3← X1 ÷ X2

X4←  DistrictName (District ⋈ X3)

1. *select DistrctName from District where DistrictID not in (select DistrictID from Stock where CurrentAmount=0)*
   * 1. Candidate keys: A
     2. Candidate keys: AB, BC, CD, AD
     3. BC → D violates BCNF since BC does not contain a key. So we split up R as in: BCD, ABC.
     4. C → A and D → B both cause violations. So decompose into: AC, BCD but this does not preserve AB → C and AB → D, and BCD is still not BCNF because D → B. So we need to decompose further into: AC, BD, CD. However, when we attempt to revive the lost functional dependencies by adding ABC and ABD, we that these relations are not in BCNF form. Therefore, there is no BCNF decomposition.