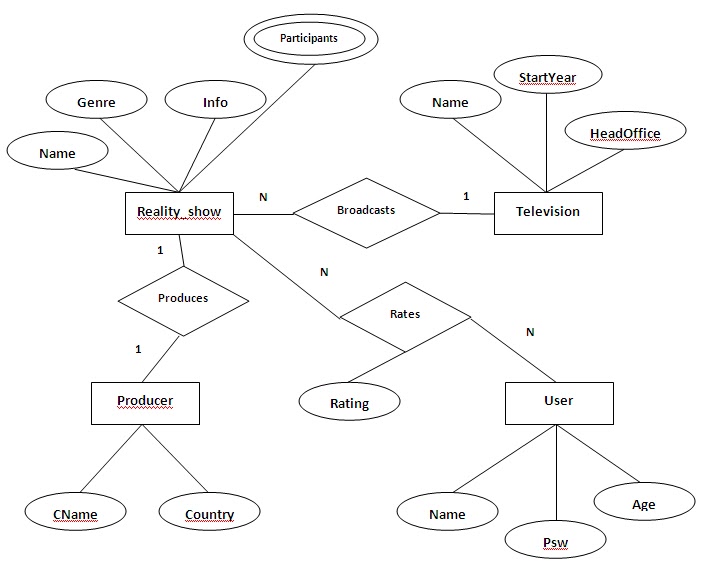
Solution 1:



Solution 2:

**Strong entity sets**

At first, let us identify and reduce all the strong entity sets (those that have primary keys) into relational schemas.

In the given diagram, ATM, Bank, Transaction, Customer, Branch, and Account are strong entity sets. To do this, we can use the name of the entity sets as schema names and all the attributes as part of the schema. In this way, we have the following schemas;

ATM (ATM\_id, cash\_limit, location)

Bank (B\_id, B\_name, B\_add)

Branch (Br\_id, Br\_name, Br\_add)

Transaction (Tr\_id, Tr\_type)

Customer (Cust\_id, Cust\_name, Cust\_add, Ph\_no)

Account (Acc\_no, Acc\_type, Balance)

Primary keys are underlined

**Composite attributes**

In entity sets Branch and Customer, the attributes Br\_add and Cust\_add respectively are composite attributes. To reduce the composite attributes we retain the component attributes in the schema. That is, instead of Br\_add, we shall include the component attributes state, country and pin the Branch table as follows;

Branch (Br\_id, Br\_name, State, Country, Pin)

Likewise, Customer becomes;

Customer (Cust\_id, Cust\_name, State, Country, Pin, Ph\_no)

**Multi-valued attributes**

Ph\_no attribute of Customer entity set is a multi-valued attribute. That is, it can have one or more values in it per record (row). To reduce the multi\_valued attribute, we need to create a separate table with a new name with the multi\_valued attribute as one attribute along with the primary key of the base entity set as follows;

Customer\_Phone (Cust\_id, Ph\_no)

The primary key for this relation is both (cust\_id, ph\_no).

**Relationship sets**

We have the following type of relationships in the given ER diagram;

Belongs – a one-to-many relationship from Bank to ATM. One bank can have many ATMs (and an ATM can belong to at most one Bank).

Has – a one-to-many relationship from Bank to Branch. One bank can have many branches (and a branch can belong to only one bank).

Operates – a many-to-many relationship from ATM to Customer. A customer can operate many ATMs and an ATM can be operated by many customers.

Performs – a many-to-one relationship from Transaction to Customer. A transaction can be performed by only one customer (and a customer can perform any number of transactions).

Holds – a many-to-many relationship from Customer to Account. A customer can have any number of accounts and an account can be maintained by more than one customer as joint accounts.

------For a many-to-many relationship, we need to create a separate schema by including the primary keys of participating entity sets as attributes.

-------For all others, the primary key of one side has to be included as the foreign key of the other side (no need to create a separate table).

Hence, we have the following changes;

Operates (ATM\_id, Cust\_id) - many-to-many

Holds (Cust\_id, Acc\_no) – many-to-many

For Belongs, include B\_id (primary key of Bank, one side) in ATM entity set (many side).

ATM (ATM\_id, cash\_limit, location, B\_id)

For Has, include B\_id (primary key of Bank, one side) in Branch entity set (many side).

Branch (Br\_id, Br\_name, State, Country, Pin, B\_id)

For Performs, include Cust\_id (primary key of Customer, one side) in Transaction entity set (many side).

Transaction (Tr\_id, Tr\_type, Cust\_id).

At the end, after reduction and modification, we have the following set of relation schemas for the given ER diagram;

Bank (B\_id, B\_name, B\_add)

Customer (Cust\_id, Cust\_name, State, Country, Pin)

Customer\_Phone(Cust\_id, Ph\_no)  
Account (Acc\_no, Acc\_type, Balance)

ATM (ATM\_id, cash\_limit, location, B\_id)

Branch (Br\_id, Br\_name, State, Country, Pin, B\_id)

Transaction (Tr\_id, Tr\_type, Cust\_id)

Operates (ATM\_id, Cust\_id), Holds (Cust\_id, Acc\_no)