Design-Your-Own Database: ‘Wine Cellar Tracker’

Roxanne Lai

CMPT 308: Database Management

Professor Schwartz

November 25, 2019

Table of Contents:

Description of what I am Modeling:

For my semester-long project I decided to model a database where users can track their wine cellar inventory along with important information they might want to know about the bottles they own. The main reason I chose to create a wine cellar tracker was due to my parent's issues with their current wine cellar setup. They always comment about how it is inconvenient to go down to their cellar without knowing what specific bottles are in there. Mainly, they pointed out that it is an issue of expectations, saying that if they have no idea what bottles are stored in their cellar, then they will have no idea what they can choose from for any given occasion. I thought that it would be useful and convenient to model a database that would keep track of wine inventory, because it is something that has a function that people can actually use in their day-to-day lives. This system will fix issues with expectations and uncertainty, because users can check what bottles they have and exactly where in their cellar those bottles are stored. The database also has information about who produced those bottles, who shipped them, which vineyard they came from, along with lots of other useful information. Due to the fact that this database can house all of this detailed information, it will lessen the work the users would have had to do to find this information. Instead of scouring the internet to find the origins of every bottle, this database has all that information in one place (if that information is available). The goal of this system is to help users keep track of their wine cellar inventories, and get rid of the uncertainty of knowing what one has in storage. Another main functionality of this database is to help users find new bottles they might want to drink for a given occasion (e.g., dinner tonight!), via the per-bottle wine review data in the database. Overall, this system will help with lowering uncertainty, find details about the bottles stored in their cellars, and makes it easy to find which bottles users want to drink based on professional reviews.

Business Rules:

1. Each wine is from only zero or one vineyard of origin.
2. A producer can produce many wines, from many vineyards
3. Each wine has zero or one appellations.
4. A wine is produced by only one producer.
5. A reviewer can review many wines.
6. A wine can be reviewed by zero or more reviewers.
7. A reviewer works for at most one company.
8. A cellar can have multiple storage locations (specific wine racks, boxes).
9. Bottles of a specific wine can be stored in multiple locations in the cellar.
10. Each wine was purchased from a single store, which can have sold multiple wines.
11. A wine or producer can be awarded zero or more awards.

ERD Diagram

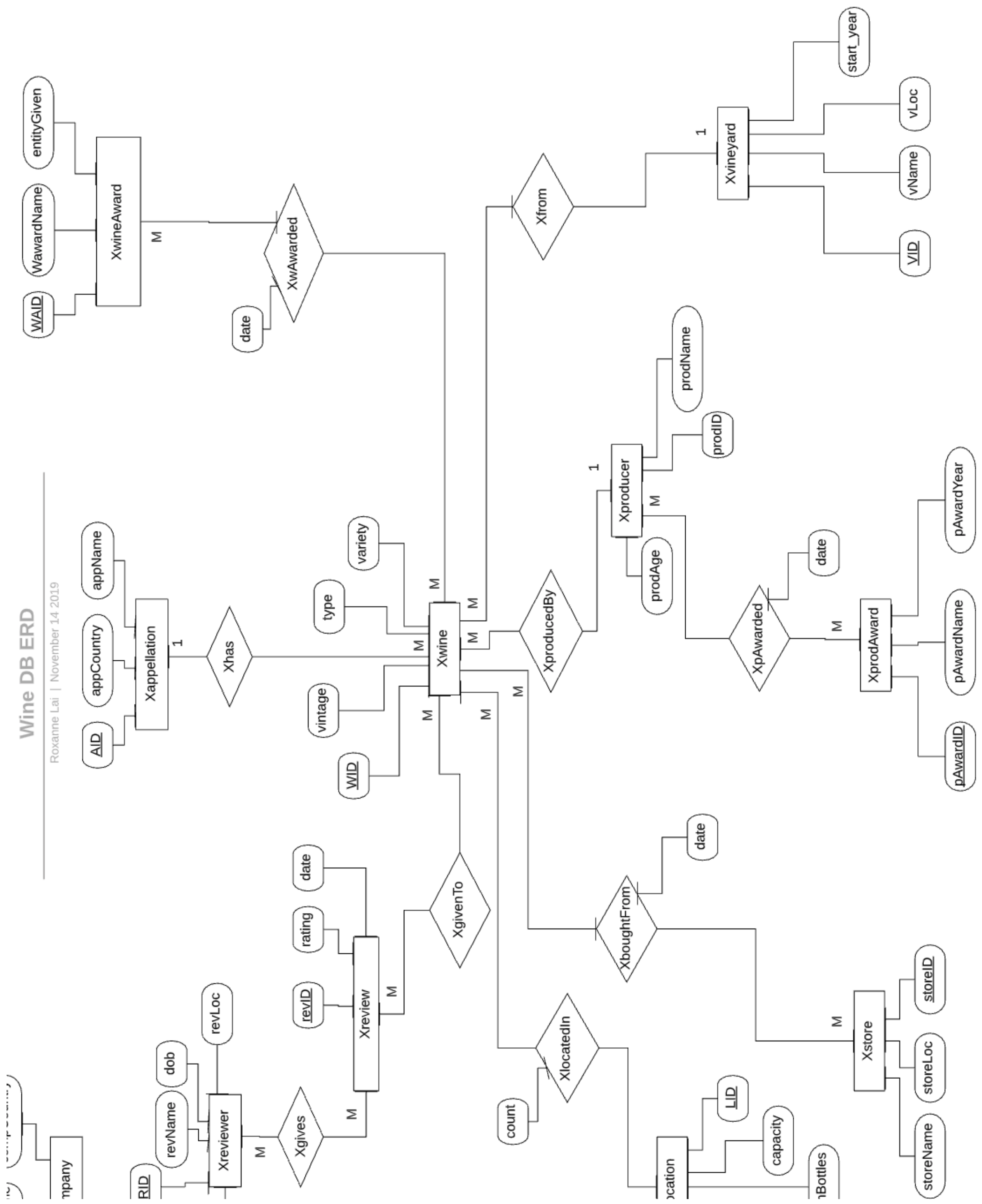

Wine DB ERD
Roxanne Lai | November 14 2019
Xwine
WID
vintage
type
variety
Xappellation
AID
appCountry
appName
Xhas
XwineAward
WAID
WawardName
entityGiven
XwAwarded
date
Xfrom
Xvineyard
VID
vName
vLoc
start_year
XproducedBy
Xproducer
prodID
prodName
prodAge
XpAwarded
date
XprodAward
pAwardID
pAwardName
pAwardYear
XboughtFrom
date
Xstore
storeID
storeLoc
storeName
XlocatedIn
count
ocation
LID
capacity
nBottles
XgivenTo
Xreview
revID
rating
date
Xgives
Xreviewer
RID
revName
dob
revLoc
mpany
1
M

Table Definitions (In Alphabetical Order):

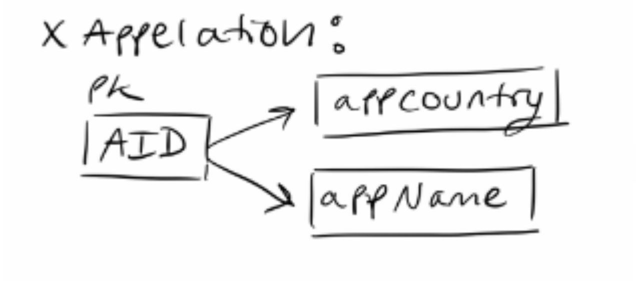
1) Xappellation table

```
CREATE TABLE Xappellation(
AID Integer,
APPCOUNTRY NVARCHAR2(30),
APPNAME NVARCHAR2(30),
CONSTRAINT PK_APPELLATION PRIMARY KEY (AID)
) ;
```

Column Name	Data Type	Description
AID	Integer	Primary key for Xappellation table, a unique number to identify each appellation
appCountry	Nvarchar2(30)	The country where the appellation is located
appName	Nvarchar2(30)	The name of the appellation

This table is the Xappellation table, some wines have an extra description, which is an appellation. For example, a 1996 Château d'Yquem's appellation is Sauternes. Some wines have an appellation descriptor and some wines do not. However, a wine can only have at most one appellation, or none. The appellation is based in a certain country and has a name as well. For example, the Sauternes are from the Sauternais region of the Graves section in Bordeaux, France. This is a long location description, so I have shortened it to be just the country, so for Sauternes the appCountry would just be France.

→ Because Xhas is a m:1 relationship, Xwine gets the fk to Xappelation(AID)



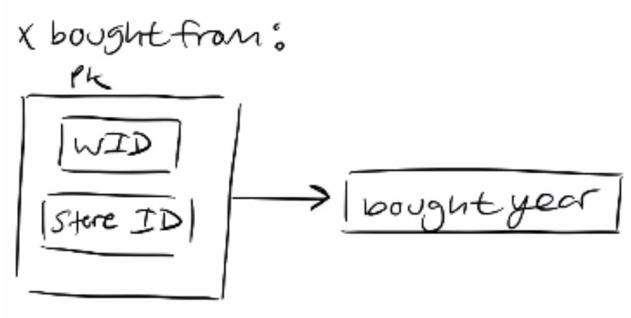
1NF: has a key, no repeating groups
2NF: is in 1NF, no partial dependencies
3NF: is in 2NF, no transitive dependencies

2) <u>XboughtFrom</u>

```
CREATE TABLE XboughtFrom(
storeID Integer,
WID Integer,
boughtYear Integer,
CONSTRAINT PK_boughtFrom PRIMARY KEY (storeID, WID)
CONSTRAINT boughtFrom_FK1 FOREIGN KEY (storeID) REFERENCES Xstore (storeID),
CONSTRAINT boughtFrom_FK2 FOREIGN KEY (WID) REFERENCES Xwine (WID)
);
```

Column Name	Data Type	Description
storeID	Integer	Foreign Key for XboughtFrom table, a unique number to identify each store.
WID	Integer	Foreign Key from the Xwine table, a unique number to identify each wine.
boughtYear	Integer	The year the wine was purchased from the store.

This table is the XboughtFrom table. It has a composite primary key (storeID, WID). The table has a storeID, which is a foreign key from the Xstore table, which is used to uniquely identify the store. It also has a foreign key from the Xwine table (WID) which shows which wine was purchased from the given store. The table also has a column for the year the wine was purchased, in case the user is interested to know what year they bought the wine. This table also has a composite primary key made up of storeID and WID. It is a M:M, so it becomes a table with fks from both tables (they become composite pk).



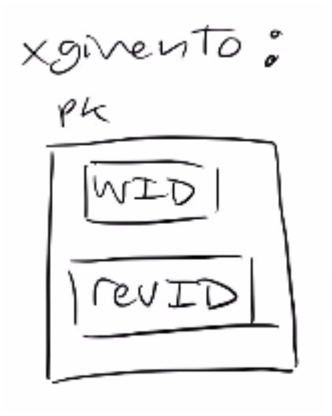
1NF: has a key, no repeating groups
2NF: is in 1NF, no partial dependencies
3NF: is in 2NF, no transitive dependencies

3) XgivenTo

```
CREATE TABLE XgivenTo(
WID Integer,
revID Integer,
CONSTRAINT XgivenTo_pk PRIMARY KEY (revID, WID)
CONSTRAINT XgivenTo_fk1FOREIGN KEY (revID) REFERENCES Xreview (revID)
CONSTRAINT XgivenT0_fk2 FOREIGN KEY (WID) REFERENCES Xwine (WID)
);
```

Column Name	Data Type	Description
WID	Integer	Foreign key, a unique number to identify each review.
revID	Integer	Foreign key, a unique number to identify each review.

This is the XgivenTo table which shows which reviews were given to which wines. It has a composite primary key (revID, WID). This table also has a foreign key from the Xwine table (WID) which tells which wine is being given the review. This table has a foreign key from the Xreview table (revID) which uniquely identifies each review by number. It is a M:M, so it becomes a table with fks from both tables (they become composite pk).



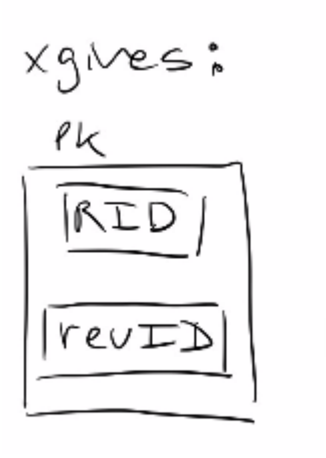
1NF: has a key, no repeating groups
2NF: is in 1NF, no partial dependencies
3NF: is in 2NF, no transitive dependencies

4) Xgives

```
CREATE TABLE Xgives(
RID Integer,
revID Integer,
CONSTRAINT Xgives_pk PRIMARY KEY (revID, RID)
CONSTRAINT Xgives_fk1FOREIGN KEY (revID) REFERENCES Xreview (revID)
CONSTRAINT Xgives_fk2 FOREIGN KEY (RID) REFERENCES Xreviewer (RID)
);
```

Column Name	Data Type	Description
RID	Integer	Foreign key, a unique number to identify each reviewer.
revID	Integer	Foreign key, a unique number to identify each review.

This is the Xgives table, which shows which reviewer gave which review(s). It has a composite primary key (revID, RID). This table has a foreign key from the Xreview table (revID) which uniquely identifies each review by number. This table has a foreign key from the reviewer table (RID), which uniquely identifies each reviewer by number. It is a M:M, so it becomes a table with fks from both tables (they become composite pk).



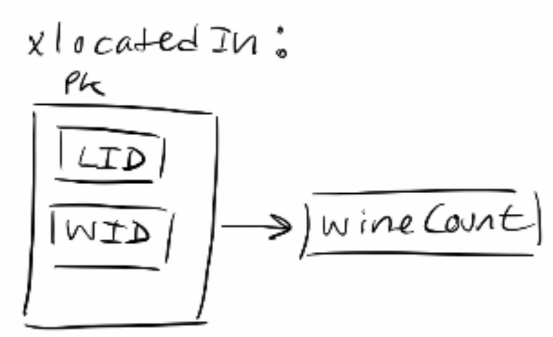
1NF: has a key, no repeating groups
2NF: is in 1NF, no partial dependencies
3NF: is in 2NF, no transitive dependencies

5) <u>XlocatedIn</u>

```
CREATE TABLE XlocatedIn(
LID Integer,
WID Integer,
wineCount Integer,
CONSTRAINT XlocatedIn_pk PRIMARY KEY (LID, WID)
CONSTRAINT locIn_FK1 FOREIGN KEY (LID) REFERENCES Xlocation (LID),
CONSTRAINT locIn_FK2 FOREIGN KEY (WID) REFERENCES Xwine (WID)
);
```

Column Name	Data Type	Description
LID	Integer	Foreign Key from the Xlocation table, a unique number to identify each location.
WID	Integer	Foreign Key from the Xwine table, a unique number to identify each wine.
wineCount	Integer	Used to keep track of how much wine is in each location.

This is the XlocatedIn table. It has a composite primary key (LID, WID)This table has a foreign key from the Xlocation table (LID), and a fk from Xwine(WID) which tells which wine is located in said location. The wineCount column is used to see how much of each wine is stored in a given location. This table also has a composite primary key made up of LID and WID. It is a M:M, so it becomes a table with fks from both tables (they become composite pk).



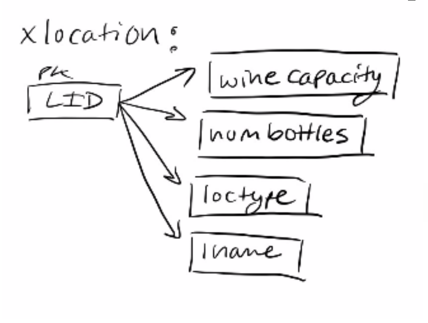
1NF: has a key, no repeating groups
2NF: is in 1NF, no partial dependencies
3NF: is in 2NF, no transitive dependencies

6) <u>Xlocation</u>

```
CREATE TABLE Xlocation(
LID Integer,
wineCapacity Integer,
numBottles Integer,
locType NVARCHAR2(30),
lName NVARCHAR2(30),
CONSTRAINT location_pk PRIMARY KEY (LID)
 );
```

Column Name	Data Type	Description
LID	Integer	Primary Key, a unique number to identify each location.
wineCapacity	Integer	Tells you how the max number of wine each location can store
numBottles	Integer	Tells you the number of bottles currently are in each location
locType	Nvarchar2(30)	Tells you what type of storage the location is (box, rack, etc.)
lName	Nvarchar2(30)	Tells you the name of the location (ex. Rack 1)

This is the Xlocation table, which gives details about the locations that bottles can be stored in. This table has a primary key (LID) which is used to uniquely identify each location. This table also has a column called wineCapacity which lets the user know the max number of wine they can store in each of the locations in their cellar. This table also has a numBottles column which shows how much wine is already in each location. The location type and name are also columns in this table to help identify where the wine is located.



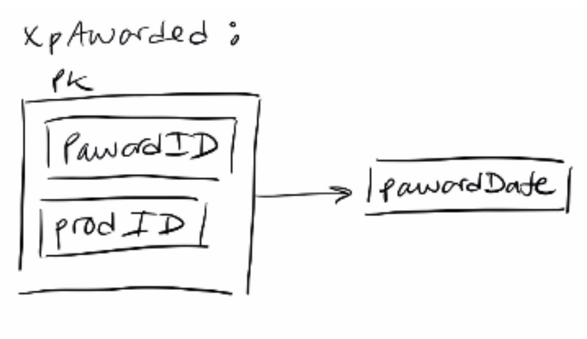
1NF: has a key, no repeating groups
2NF: is in 1NF, no partial dependencies
3NF: is in 2NF, no transitive dependencies

7) XpAwarded

```
CREATE TABLE XpAwarded (
pAwardID Integer,
prodIDInteger,
pAwardDate Integer,
 CONSTRAINT pAwarded_fk1  FOREIGN KEY (pAwardID) REFERENCES XprodAward
(pAwardID),
CONSTRAINT pAwarded_fk2 FOREIGN KEY (prodID) REFERENCES Xproducer (prodID)
);
```

Column Name	Data Type	Description
pAwardID	Integer	This is a foreign key, a unique number to identify each producer award.
prodID	Integer	This is a foreign key, a unique number to identify producer.
pAwardDate	Integer	This is the date that the award was given out (just the year)

This is the XpAwarded table, which shows which awards were given to which producers. It has a foreign key from the XprodAward table (pAwardID), which shows which award was given out. It also has a foreign key from the Xproducer table (prodID), which shows which producer the award was given to. This table also has a column called pAwardDate, which shows what year the award was given. It is a M:M, so it becomes a table with fks from both tables (they become composite pk).



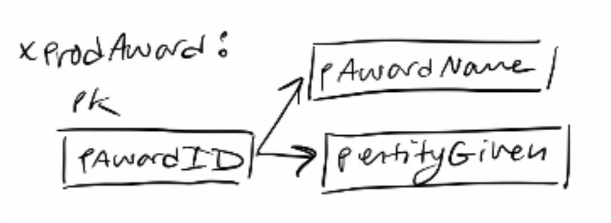
1NF: has a key, no repeating groups
2NF: is in 1NF, no partial dependencies
3NF: is in 2NF, no transitive dependencies

8) <u>XprodAward</u>

```
CREATE TABLE XprodAward(
pAwardID Integer,
pAwardName Nvarchar2(30),
pEntityGiven Nvarchar2(30),
 CONSTRAINTprodAward_pk  PRIMARY KEY (pAwardID)
);
```

Column Name	Data Type	Description
pAwardID	Integer	This is the primary key, a unique number to identify each producer award.
pAwardName	Nvarchar2(30)	This is the name of the award
pEntityGiven	Nvarchar2(30)	The entity that gave the award.

This is the XprodAward table, which describes the awards that can be given to producers. This table has a primary key (pAwardID) which is a unique number assigned to each award. This table also has a column for the award's name. There is also a pEntityGiven column which shows which entity/group gave out the wine award.



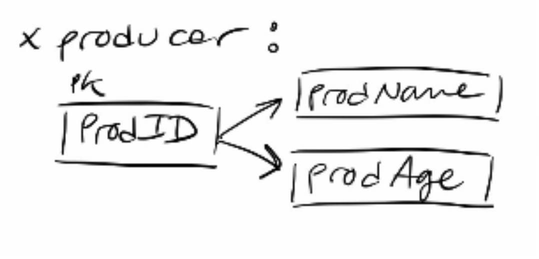
1NF: has a key, no repeating groups
2NF: is in 1NF, no partial dependencies
3NF: is in 2NF, no transitive dependencies

9) Xproducer

```
CREATE TABLE Xproducer(
prodID Integer,
prodName Nvarchar2(30),
prodAge Integer,
 CONSTRAINT producer_pk  PRIMARY KEY (prodID)
);
```

Column Name	Data Type	Description
prodID	Integer	This is the primary key, a unique number to identify each producer.
prodName	Nvarchar2(30)	This is the name of the producer.
prodAge	Integer	This is the year the producer was born.

This is the Xproducer table, which describes the producers This table has a primary key (prodID) which is a unique number assigned to each producer. This table also has a column for the producer's name. There is also a prodAge column which shows the year the producer was born.



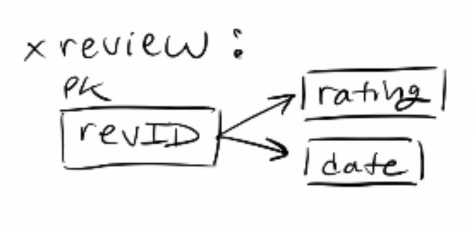
1NF: has a key, no repeating groups
2NF: is in 1NF, no partial dependencies
3NF: is in 2NF, no transitive dependencies

10) <u>Xreview</u>

```
CREATE TABLE Xreview(
revID Integer,
rating Integer,
revYear Integer,
CONSTRAINT review_pk PRIMARY KEY (revID)
);
```

Column Name	Data Type	Description
revID	Integer	Primary key, a unique number to identify each review.
rating	Integer	The rating that the review gives the wine (out of 100)
revYear	Integer	The year the review was given

This is the Xreview table, which shows which review it was, the reviewer, and the wine the review was given to. This table has a primary key (revID) which uniquely identifies each review by number. In addition this table also has a rating and a revYear column which specify the rating the review gave to the wine, and the year that the review was given.



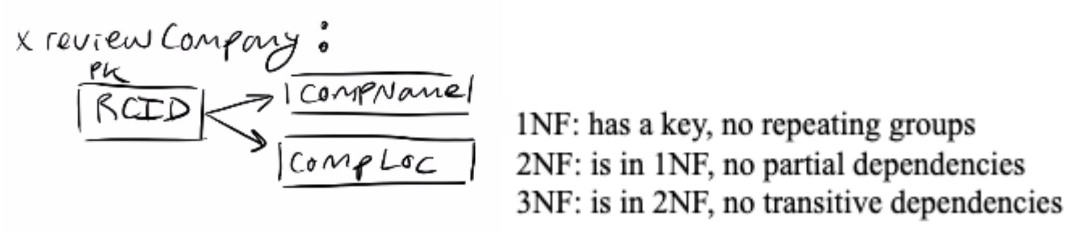
1NF: has a key, no repeating groups
2NF: is in 1NF, no partial dependencies
3NF: is in 2NF, no transitive dependencies

11) XreviewCompany

```
CREATE TABLE Xreview(
RCID Integer,
compName Nvarchar2(30),
compCountry Nvarchar2(30),
CONSTRAINT reviewCompany_pk PRIMARY KEY (RCID)
);
```

Column Name	Data Type	Description
RCID	Integer	Primary key, a unique number to identify each review company.
compName	Nvarchar2(30)	The name of the review company
compCountry	Nvarchar2(30)	The country the review company is based in

This is the XreviewCompany table, which gives details about a review company. This table has a primary key (RCID) which uniquely identifies each review company by number. This table also has a column called compName which gives the company name for each review company. This table also has a column called compCountry which gives the country name where the review company is located in.



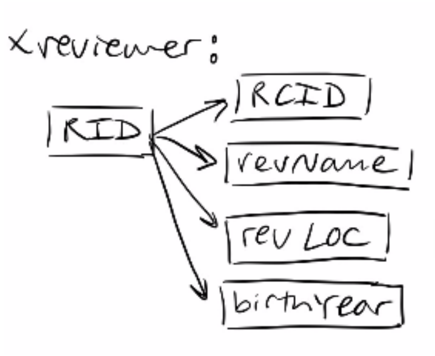
1NF: has a key, no repeating groups
2NF: is in 1NF, no partial dependencies
3NF: is in 2NF, no transitive dependencies

12) Xreviewer

```
CREATE TABLE Xreview(
RID Integer,
RCID Integer,
revName Nvarchar2(30),
revLoc Nvarchar2(30),
birthYear Integer,
CONSTRAINT reviewer_pk PRIMARY KEY (RID)
);
```

Column Name	Data Type	Description
RID	Integer	Primary key, a unique number to identify each reviewer.
RCID	Integer	Foreign key, a unique number to identify each review company.
revName	Nvarchar2(30)	The name of the reviewer
revLoc	Nvarchar2(30)	The country the reviewer is located in.
birthYear	Integer	The year the reviewer was born.

This is the Xreviewer table, which gives information about each reviewer. This table has a primary key (RID), which uniquely identifies each reviewer by number. This table also has a foreign key(RCID). There is also a revName column which gives the reviewer's name, a revLoc column which shows the country the reviewer is based in, and a birthYear column which gives the birth year of the reviewer.→ Because XworksFor is a m:1 relationship, Xreviewer gets the fk to XreviewCompany(RCID)



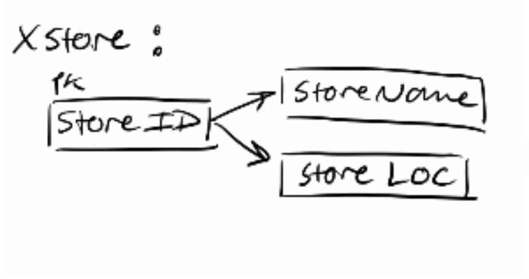
1NF: has a key, no repeating groups
2NF: is in 1NF, no partial dependencies
3NF: is in 2NF, no transitive dependencies

13) Xstore

```
CREATE TABLE Xstore(
storeID Integer,
storeName Nvarchar2(30),
storeLoc Nvarchar2(30),
CONSTRAINT store_pk PRIMARY KEY (storeID)
);
```

Column Name	Data Type	Description
storeID	Integer	Primary key, a unique number to identify each store.
storeName	Nvarchar2(30)	The name of the store
storeLoc	Nvarchar2(30)	The country the store is located in.

This is the Xstore table, which gives information about each store. This table has a primary key (storeID), which uniquely identifies each store by number. There is a storeName column which gives the store's name, and a storeLoc column which shows the country the store is located in.



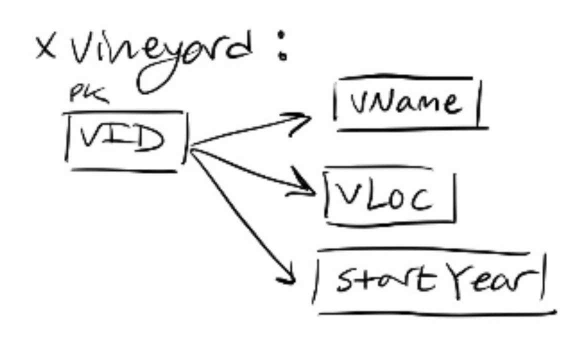
1NF: has a key, no repeating groups
2NF: is in 1NF, no partial dependencies
3NF: is in 2NF, no transitive dependencies

14) Xvineyard

```
CREATE TABLE Xvineyard(
VID Integer,
vName Nvarchar2(30),
vLoc Nvarchar2(30),
start_year Integer,
CONSTRAINT vineyard_pk PRIMARY KEY (VID)
);
```

Column Name	Data Type	Description
VID	Integer	Primary key, a unique number to identify each vineyard.
vName	Nvarchar2(30)	The name of the vineyard.
vLoc	Nvarchar2(30)	The country the vineyard is located in.
start_year	Integer	The year the vineyard was founded.

This is the Xvineyard table, which gives information about each vineyard. This table has a primary key (VID), which uniquely identifies each vineyard by number. There is a vName column which gives the vineyard's name. There is also a vLoc column which shows the country the vineyard is located in. The table also has a start_year column which gives the year the vineyard was founded.



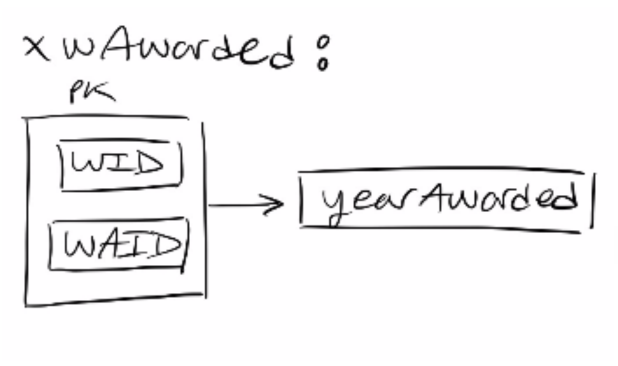
1NF: has a key, no repeating groups
2NF: is in 1NF, no partial dependencies
3NF: is in 2NF, no transitive dependencies

15) XwAwarded

```
CREATE TABLE WxAwarded(
WAID Integer,
WID Integer,
yearAwarded Integer,
CONSTRAINT XwAwarded_fk1 FOREIGN KEY (WID) REFERENCES XwineAward
(WAID),
CONSTRAINT XwAwarded_fk2 FOREIGN KEY (WID) REFERENCES Xwine (WID)
);
```

Column Name	Data Type	Description
WAID	Integer	Foreign key, a unique number to identify each wine award.
WID	Integer	Foreign key, a unique number to identify each wine.
yearAwarded	Integer	The year the award was given out.

This is the XwAwarded table, which gives information about each wine that was awarded. This table has a foreign key from the XwineAward table(WAID), which uniquely identifies each wine award by number. This table also has a foreign key from the Xwine table(WID), which uniquely identifies each wine the award was given to (if the award was given out). There is also a yearAwarded column which gives the year the award was given out (if it was given out that year). It is a M:M, so it becomes a table with fks from both tables (they become composite pk).



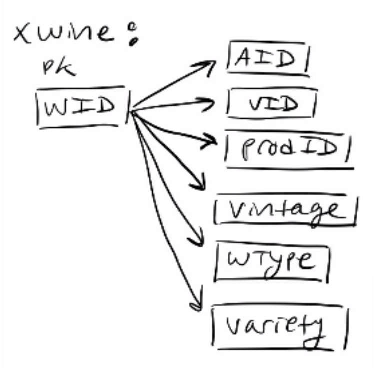
1NF: has a key, no repeating groups
2NF: is in 1NF, no partial dependencies
3NF: is in 2NF, no transitive dependencies

16) Xwine

```
CREATE TABLE Xwine(
WID Integer,
AID Integer,
vintage Nvarchar2(30),
wType Nvarchar2(30),
Variety Nvarchar2(30),
VID  Integer,
yearAwarded Integer,
CONSTRAINT wine_pk PRIMARY KEY (WID),
CONSTRAINT wine_fk1 FOREIGN KEY (AID) REFERENCES Xappellation (AID),
CONSTRAINT wine_fk2 FOREIGN KEY (VID) REFERENCES Xvineyard (VID)
CONSTRAINT wine_fk3 FOREIGN KEY (prodID) REFERENCES Xproducer(prodID)
);
```

Column Name	Data Type	Description
WID	Integer	Primary key, a unique number to identify each wine.
AID	Integer	Foreign key, a unique number to identify each appellation.
vintage	Nvarchar2(30)	The vintage of the wine
wType	Nvarchar2(30)	The type of wine
variety	Nvarchar2(30)	The variety of the wine
VID	Integer	Foreign key, a unique number to identify each vineyard.
ProdID	Integer	Foreign key, a unique number to identify each producer.

This is the Xwine table, which has a primary key (WID) that uniquely identifies each wine award by number. This table has a fk from the Xappellation table, a fk from the Xvineyard table, and a fk from the Xproducer table. There is a vintage column which gives the vintage of the wine, a wType column which gives the type of the wine, and a variety column which gives the variety of the wine.→ Because XproducedBy is a m:1 relationship, Xwine gets the fk to Xproducer(prodID). Xfrom is also m:1 so Xwine gets the fk to Xvineyard(VID).



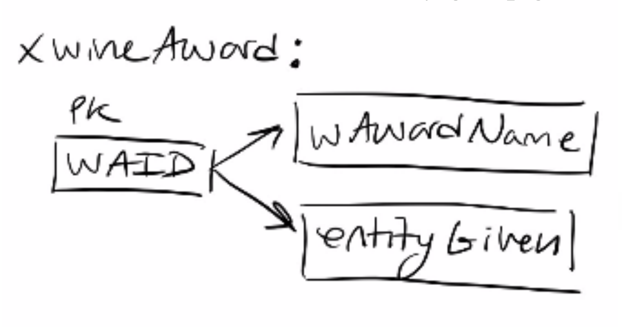
1NF: has a key, no repeating groups
2NF: is in 1NF, no partial dependencies
3NF: is in 2NF, no transitive dependencies

17) XwineAward

```
CREATE TABLE XwineAward(
WAID Integer,
wAwardName Nvarchar2(30),
entityGiven Nvarchar2(30),
CONSTRAINT XwineAwarded_pk PRIMARY KEY (WAID)
);
```

Column Name	Data Type	Description
WAID	Integer	Primary key, a unique number to identify each wine award.
wAwardName	Nvarchar2(30)	The name of the award
entityGiven	Nvarchar2(30)	The entity that gave the award.

This is the XwineAward table, which gives information about each wine award. This table has a primary key (WAID), which uniquely identifies each wine award by number. There is a wAwardName column which gives the name of the wine award. There is also an entityGiven column which shows which entity/group gave out the wine award.



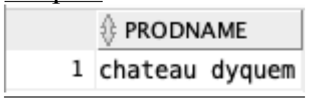
1NF: has a key, no repeating groups
2NF: is in 1NF, no partial dependencies
3NF: is in 2NF, no transitive dependencies

Queries:

1) “Every/All” query

<u>English:</u>
“Name producers for whom there is not a produce- award they haven’t won. AKA name producers who have won all the producer awards.”

<u>Query:</u>
```
Select xproducer.prodName
From xproducer
Where not exists
        (select *
        From xprodaward
        Where not exists
                (select *
                From xpawarded
                Where xprodaward.pawardID = xpawarded.pawardID
                And xpawarded.prodID = xproducer.prodID)));
```

<u>Output:</u>

```
     PRODNAME
  1  chateau dyquem
```

<u>Cardinality:</u>
Cardinality = 1

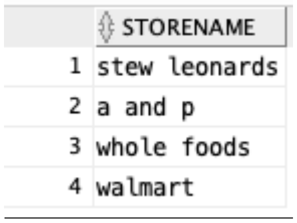
2) “Only” query

English:
“Name stores that have only sold red wine”

Query:

```
Select xstore.storeName
From xstore
Where xstore.storeID not in
        (select xboughtfrom.storeID
        From xboughtfrom
        Where xboughtfrom.WID not in
                (select xwine.WID
                From xwine
                Where xwine.wType = 'red'));
```

Output:



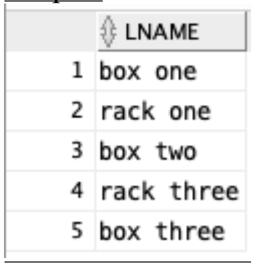
	STORENAME
1	stew leonards
2	a and p
3	whole foods
4	walmart

Cardinality:
Cardinality = 4

3) “None” query

English:
“Name locations that store none of the white wines”

Query:
```
Select xlocation.LName
From xlocation
Where xlocation.LID not in
        (select xlocatedIn.LID
        From xlocatedIn
        Where xlocatedIn.WID not in
                (select xwine.WID
                From xwine
                Where xwine.wType <> 'white'));
```

Output:


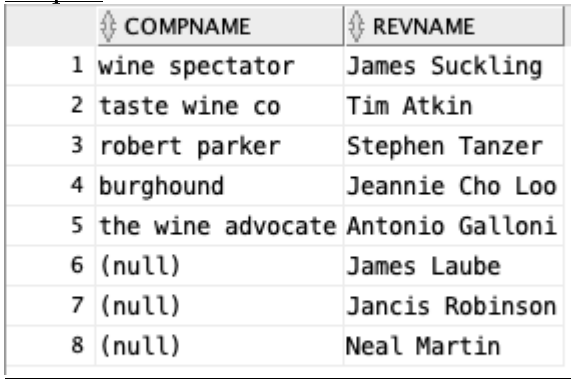
	LNAME
1	box one
2	rack one
3	box two
4	rack three
5	box three

Cardinality:
Cardinality = 5

4) “Right Outer Join” query

English:
“Name reviewers and the review-company they work (for if any) where the reviewer is not from France.”

Query:
select XreviewCompany.compName, Xreviewer.revName
FROM XreviewCompany RIGHT JOIN Xreviewer ON Xreviewer.RCID = XreviewCompany.RCID
WHERE xreviewer.revLoc <> 'France';

Output:


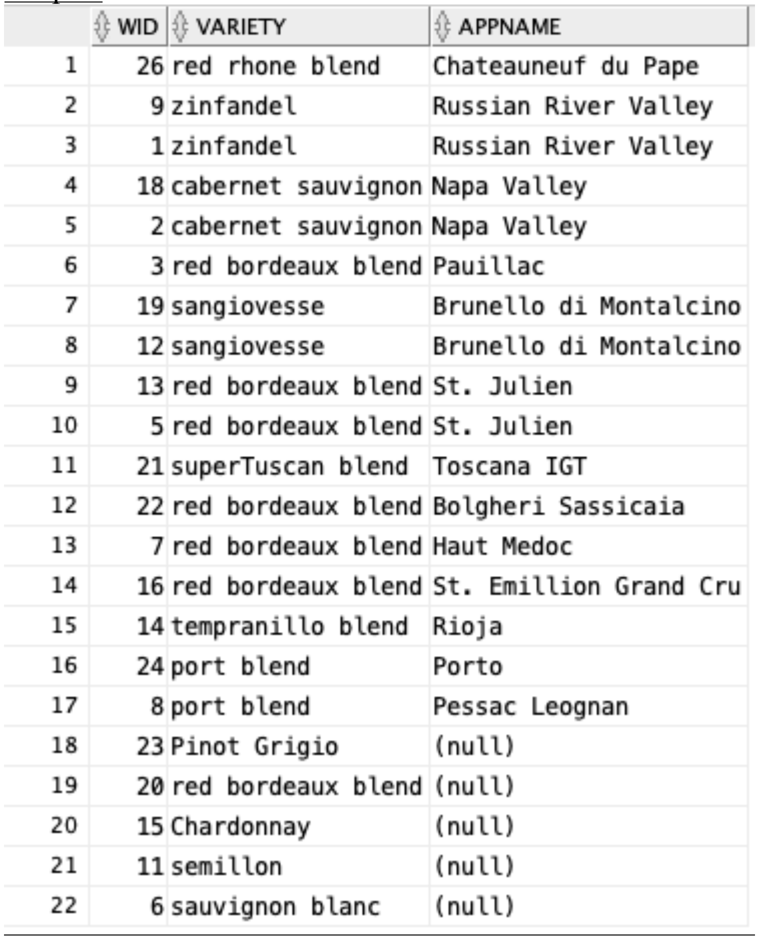
	COMPNAME	REVNAME
1	wine spectator	James Suckling
2	taste wine co	Tim Atkin
3	robert parker	Stephen Tanzer
4	burghound	Jeannie Cho Loo
5	the wine advocate	Antonio Galloni
6	(null)	James Laube
7	(null)	Jancis Robinson
8	(null)	Neal Martin

Cardinality:
Cardinality = 8

5) "Left Outer Join" query

English:
"Give the ID number and variety for wine, and the name of their appellation if any"

Query:
```
select Xwine.WID, xwine.variety, Xappellation.appName
FROM Xwine LEFT JOIN Xappellation ON Xwine.AID = Xappellation.AID
WHERE xwine.vintage <= 2017;
```

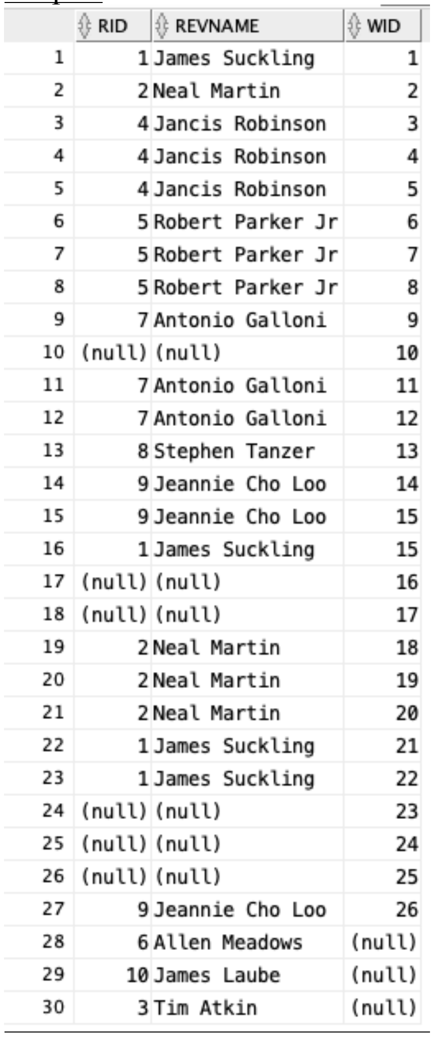
Output:

```
      WID  VARIETY             APPNAME
 1     26  red rhone blend     Chateauneuf du Pape
 2      9  zinfandel           Russian River Valley
 3      1  zinfandel           Russian River Valley
 4     18  cabernet sauvignon  Napa Valley
 5      2  cabernet sauvignon  Napa Valley
 6      3  red bordeaux blend  Pauillac
 7     19  sangiovesse         Brunello di Montalcino
 8     12  sangiovesse         Brunello di Montalcino
 9     13  red bordeaux blend  St. Julien
10      5  red bordeaux blend  St. Julien
11     21  superTuscan blend   Toscana IGT
12     22  red bordeaux blend  Bolgheri Sassicaia
13      7  red bordeaux blend  Haut Medoc
14     16  red bordeaux blend  St. Emillion Grand Cru
15     14  tempranillo blend   Rioja
16     24  port blend          Porto
17      8  port blend          Pessac Leognan
18     23  Pinot Grigio        (null)
19     20  red bordeaux blend  (null)
20     15  Chardonnay          (null)
21     11  semillon            (null)
22      6  sauvignon blanc     (null)
```

Cardinality:
Cardinality = 22

6) “Full Outer Join” query

English:
‘Give the name and ID numbers for reviewers that have reviewed wine (if any), and the ID numbers for wine that have reviews (if any).

Query:
```
SELECT xreviewer.RID, xwine.WID
FROM Xreviewer FULL JOIN  Xgives ON xgives.RID = Xreviewer.RID
FULL JOIN Xreview ON Xgives.revID = Xreview.revID
FULL JOIN XgivenTo ON XgivenTo.revID = Xreview.revID
FULL JOIN Xwine on XgivenTo.WID = Xwine.WID;
```

Output:


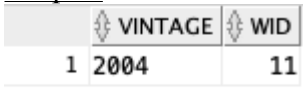
	RID	REVNAME	WID
1	1	James Suckling	1
2	2	Neal Martin	2
3	4	Jancis Robinson	3
4	4	Jancis Robinson	4
5	4	Jancis Robinson	5
6	5	Robert Parker Jr	6
7	5	Robert Parker Jr	7
8	5	Robert Parker Jr	8
9	7	Antonio Galloni	9
10	(null)	(null)	10
11	7	Antonio Galloni	11
12	7	Antonio Galloni	12
13	8	Stephen Tanzer	13
14	9	Jeannie Cho Loo	14
15	9	Jeannie Cho Loo	15
16	1	James Suckling	15
17	(null)	(null)	16
18	(null)	(null)	17
19	2	Neal Martin	18
20	2	Neal Martin	19
21	2	Neal Martin	20
22	1	James Suckling	21
23	1	James Suckling	22
24	(null)	(null)	23
25	(null)	(null)	24
26	(null)	(null)	25
27	9	Jeannie Cho Loo	26
28	6	Allen Meadows	(null)
29	10	James Laube	(null)
30	3	Tim Atkin	(null)

Cardinality:
Cardinality = 30

7) "6 Tables" query

English:
"Get the vintage and ID number for the wine where the producer was born before 1900, stored in the location with ID number greater than or equal to 5, was purchased in 2018, the producer of the wine was given an award in 2002, and the wine itself won an award in or after the year 2000"

Query:
```
Select distinct xwine.vintage, xwine.WID
From xwine, xlocatedIn, xboughtfrom, xproducer, xpawarded, xwawarded
Where xwine.WID = xlocatedIn.WID
AND xwine.WID = xboughtfrom.WID
AND xwine.prodID = xproducer.prodID
AND xpawarded.prodID = xproducer.prodID
AND xwine.WID = xwawarded.WID
AND xproducer.prodage < 1900
AND xlocatedIn.LID >= 5
AND xboughtfrom.boughtyear = 2018
AND xpawarded.pawarddate = 2002
AND xwawarded.yearawarded >= 2000;
```

Output:


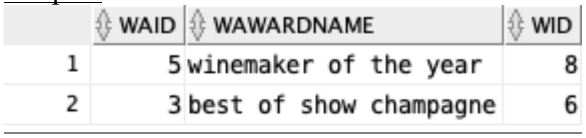
	VINTAGE	WID
1	2004	11

Cardinality:
Cardinality = 1

8) “Complex 1” query

English:
“Give the ID numbers and names of awards that were given to a wine (give the ID numbers for the wine as well) that was reviewed by Robert Parker Jr”

Query:
```
SELECT DISTINCT xwawarded.WAID, xwineaward.wawardname, xwine.WID
FROM xwawarded, xwine, xgives, xgivento, xreviewer, xreview, xwineaward
WHERE xwawarded.WID = xwine.WID
AND xwawarded.WAID = xwineaward.WAID
AND xgivento.WID = xwine.WID
AND xgivento.revID = xreview.revID
AND xgives.revID = xreview.revID
AND xgives.RID = xreviewer.RID
AND xreviewer.revname = 'Robert Parker Jr';
```

Output:


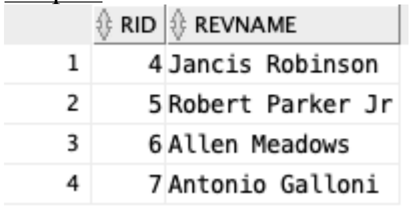
	WAID	WAWARDNAME	WID
1	5	winemaker of the year	8
2	3	best of show champagne	6

Cardinality:
Cardinality = 2

9) "Complex 2" query

English:
"Get reviewer Id numbers and names for reviewers that are younger than Stephen Tanzer"

Query:
```
SELECT DISTINCT r1.RID, r1.revname
FROM xreviewer r1
WHERE r1.birthyear <
(SELECT r2.birthyear
FROM xreviewer r2
WHERE r2.revname = 'Stephen Tanzer');
```

Output:


	RID	REVNAME
1	4	Jancis Robinson
2	5	Robert Parker Jr
3	6	Allen Meadows
4	7	Antonio Galloni

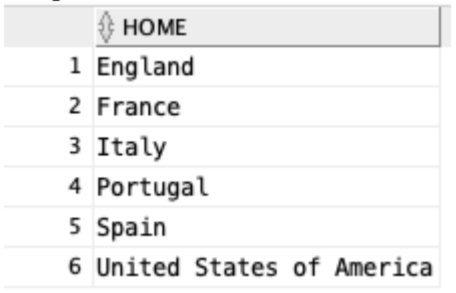
Cardinality:
Cardinality = 4

10) "Complex 3" query

English: "name the countries with which at least one vineyard, store, or appellation is associated"

Query:
```
SELECT DISTINCT xvineyard.vloc Home
From xvineyard

UNION

SELECT DISTINCT xstore.storeloc Home
FROM xstore

UNION

SELECT DISTINCT xappellation.appcountry Home
FROM xappellation;
```

Output:



```
   HOME
1  England
2  France
3  Italy
4  Portugal
5  Spain
6  United States of America
```

Cardinality:
Cardinality = 6