NAME:_______________________ NEPTUN CODE: _______________1/3

5 points/exercise; 9 excercise x 5 points = 45 points; 15p (2), 21p (3), 27p (4), 33p-45p (5)
20 minutes/page; 3 page x 20 min = 60 minutes/Exam

```
Beers(beerID, Name, Manf, Country)
Bars(barID, Name, Country)
Sells(beerID, barID, price)
```

1.) Queries in core relational algebra, core, not-extended algebra, do not use aggregation functions:
Which beers are sold in exactly one bar? Give the pair of names of (Beers.Name, Bars.Name)
You can use only core relational algebraic operators!

2.) Queries in SQL, SELECT statement: Which bars sell only beers manufactured in its own country?
Give the Names of bars! („Manufactured in its own country" means Beers.Country= Bars.Country)

3.) Queries with grouping and aggregation in SQL SELECT statement: List the main aggregation operators.
Compare WHERE and HAVING clauses. Determine the average price of beers for every bars in Japan but only for those bars where at least three beers are sold. Give the pair of (beerID, avg_price).

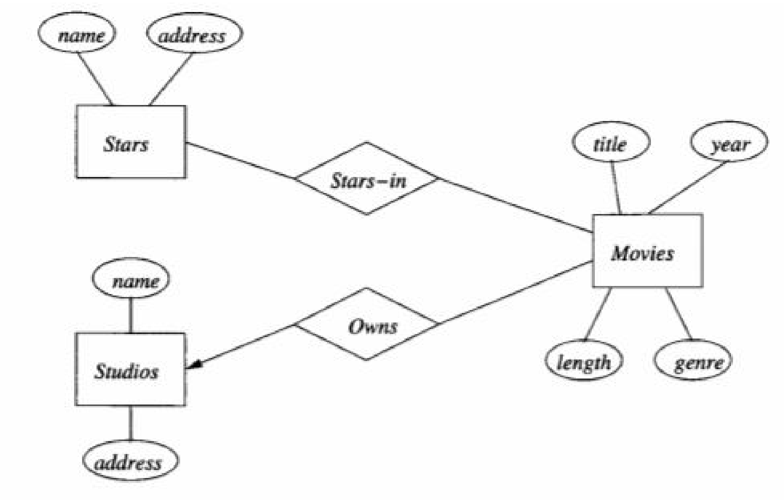
```
Beers(beerID, Name, Manf, Country)
Bars(barID, Name, Country)
Sells(beerID, barID, price)
```

4.) SQL DDL. Give the CREATE TABLE statements to define Beers, Bars and Sells tables, see above, with primary keys (the primary keys are underlined), foreign keys and a referential integrity constraint.

5.) SQL DML. List the SQL DML statements. Delete all beers for which there is another beer by the same manufacturer.

6.) Recursion in DATALOG and SQL WITH RECURSIVE: Given Flights(airline, fromcity, tocity,price). For what pairs of cities(x, y) and what price (z) is it possible to get from city x to city y by taking one or more flights?

7.) High-level Database Models: Convert this E /R diagram to a relational database schema:



8.) Design of Relational Database Schemas: Define the Boyce-Codd normal form (BCNF). Determine the keys in the example, and determine which dependencies break the BCNF, use the closure set of attributes X+ algorithm to test. Let R = ABCDE and the sets of FD's = {A -> D, AC -> E, DE -> B}.

9.) Define lossless join decomposition. Let R=ABCDE and R be decomposed into relations with the following three sets of attributes : R1=ABC, R2=BCD, R3=ACE. Use the chase test to tell whether the decomposition of R is lossless or not for the following sets of functional dependencies:

a.) F1 = {AC → E, BC → D},
b.) F2 = {A → D , D → E, B → D}.