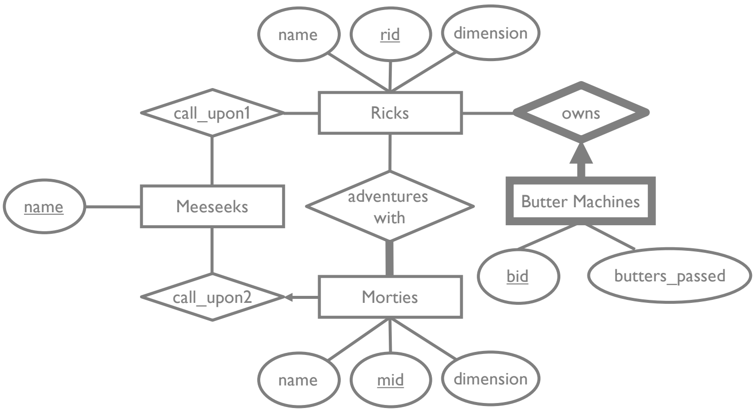
1 (18 points) Terms and Definitions

(3 points each) In *at most* two short sentences each, explain the meaning of the following terms as they relate to database management systems. **Draw a box around your final answer**.

1. **Total Participation**
2. **Transaction**
3. **Bag/Multi-Set Semantics**
4. **Positional-field Notation**
5. **User Defined Functions**
6. **Physical-data Independence**

2 (28 points) Buliting a Multi-verse using ER diagrams

Consider the following Entity-Relationship diagram that describes relationships in Rick and Morty's world. There are multiple Ricks and Morties, each from a particular dimension. Ricks go on multi-dimensional adventures with Morties. Ricks also own butter machines whose job is to pass the butter. In addition, Meeseeks are creatures that can be called upon by Ricks and/or Morties to accomplish tasks – in fact, that is their sole purpose. The description above does not describe the multiplicity of the relationships in the diagram, which is your job to understand!



1. **(12 points)** For the next 4 items, answer them based on the ER diagram above. You will get 3 points for each correct answer, -0.5 points for each incorrect answer, and 0 points for each answer left blank.

   (a) **True / False** There are Morties that grow up and live their lives without going on any adventures with a Rick.

   (b) **True / False** A single Meeseek can been called upon by both a Rick from dimension `C-137` and a Morty from dimension `C-137`.

   (c) **True / False** Ricks have a lot of enemies (including other Ricks) and might be exterminated, thus removing the dead Rick from the Ricks table. In that situation, does this allow the deceased Rick's Butter Machine to run away and live a long, prosperous, non-butter-passing life?

   (d) Consider Morty from dimension `C-137`, let's call him $Morty_{c137}$. $Morty_{c137}$ is feeling overwhelmed about missing math class to go on adventures and has called upon as many Meeseeks as he can (according to the ER diagram). Circle the most general correct answer.

   i. $Morty_{c137}$ must have called more Meeseeks than some Ricks.
   ii. $Morty_{c137}$ must have called more Meeseeks than all Ricks.
   iii. $Morty_{c137}$ must have called less Meeseeks than some Ricks.
   iv. $Morty_{c137}$ must have called less Meeseeks than all Ricks.
   v. None of the above.

2. **(16 points)** Consider the following SQL schema, which translates of the ER diagram above into tables in the relational model.

```
CREATE TABLE Ricks(
  rid int UNIQUE,
  name text,
  dimension text
);
```

```
CREATE TABLE MortiesAdventuresWith(
  mid int NOT NULL,
  name text,
  dimension text,
  rid int REFERENCES Ricks(rid),
  mname_call_upon2 text REFERENCES Meeseeks2(name),
  PRIMARY KEY (mid)
)

CREATE TABLE ButterMachines(
  bid int,
  butters_passed int,
  check (bid is not null),
  rid int REFERENCES Ricks(rid) NOT NULL ON CASCADE delete,
  PRIMARY KEY (bid, rid)
)

CREATE TABLE Meeseeks1(
  name text UNIQUE NOT NULL,
  PRIMARY KEY (name)
);

CREATE TABLE call_upon1(
  rick_id int,
  mname text REFERENCES Meeseeks1(name),
  UNIQUE (rick_id, mname)
);

CREATE TABLE Meeseeks2(
  name text UNIQUE NOT NULL,
  PRIMARY KEY (name)
);
```

(4 points each) List and briefly explain the four most important problems that you find with this schema, in terms of how well it models the E/R diagram above. For each problem, provide a *very* brief example of data that illustrates the problem. Do not base your answer on comparing this schema with other possible schemas. Instead, just compare the schema against the E/R diagram to identify what is not captured from the diagram, etc. in the schema.

You will be graded on the importance of the problems that you identify and the extent that the problems you describe overlap. For example, if you say "table Foo is wrong" as one problem, and "relationship with Foo is a problem because Foo is wrong", then we will grade the second "problem" lower.

Braw a box around your final answer.

(a) **Problem 1:**

(b) **Problem 2:**

(c) **Problem 3:**

(d) **Problem 4:**

3 (26 points) Save the Multi-verse with SQL

Consider the following intergalactic friendship database, consisting of two relations for `users` and `friends`. The direction of the friendship is explicitly stored, such that the friender `A` initiated the friendship and the friendee `B` accepts the friendship. This means `B` can also initiate a friendship with `A`, so that `B` is in two records – one as a friender and one as a friendee. Despite this, we say that users `X` and `Y` are friends regardless of which one is the friender and friendee. Users cannot be friends with themselves.

```
CREATE TABLE users(
  uid int PRIMARY KEY,
  username text UNIQUE NOT NULL,
  species text
);

CREATE TABLE friends(
  friender int REFERENCES users(uid),
  friendee int REFERENCES users(uid),
  when_added date,
  PRIMARY KEY (friender, friendee)
);
```

In the following 3 questions, write **a single** SQL statement that satisfies the english statement. You may use SQL features such as the `WITH` clause. Draw a box around your final answer. In the last question, you will fill in a `TRIGGER` template to express the english statement.

1. **(6 points)** `Warren-G` and `Ice-T` are `Alphabetrians` (users of the species = `Alphabetrian`). Find the usernames of all users that are friendees of both of these `Alphabetrians`.

2. **(6 points)** `Alphabetrians` are at war with `Numbericons` (users of the species = `Numbericon`) and it's time to *crunch the numbers*. We need to list our `Alphabetrian` brethren and note down the traitors (users that are friends with a `Numbericon`). Create a two column table containing all `Alphabetrians`. The first column is the username and the second column is `TRUE` if the user is a traitor, and `NULL` or `FALSE` otherwise. The names of the columns don't matter.

3. **(8 points)** Rick is wanted by the Inter-dimensional Federation and we're going to use social media to crack down on his rebel group. Find the usernames of all users that are frienders of all of Rick's friends (Rick has username = `Rick Sanchez`)

4. **(6 points)** Fill in the following logical `TRIGGER` definition (not postgresql) using `PL/SQL` to prevent `Cromulons` from adding themselves (users of the species = `Cromulon`) to the `users` table.

```
CREATE TRIGGER nocromulons

__________________ INSERT ON ________________________

FOR EACH ___________________________
  BEGIN
```

4 (10 points) Inter-dimensional Relations with Relational Algebra

Consider the database in Problem 3. Write the relational algebra query to find the usernames of users that are frienders with ALL users of species = `Alphabetrian`, but NOT frienders with species = `Numbericon`.

Draw a BOX around your final and intermediate answers – partial credit will be given to *clearly labeled* intermediate expressions

5 (16 points) A =? B

Consider the relational model and two relations with the same schema, R and S. You may assume that no primary keys have been explicitly defined by the database administrator, but otherwise do not make any assumptions about the database.

```
R(a int, b int)
S(a int, b int)
```

Each row of the following table shows two queries. In the blank third column of the table write YES if the two queries are equivalent, and NO if they are not equivalent. Two queries are equivalent if they always return exactly the same result relation on all databases with the schema above.

You will receive **4 points** for each correct answer, **-1 points** for each incorrect answer, **0 points** for each answer left blank.

Q1	**Q2**	**Equivalent?**
$\pi_{R.a}((R \cup S) - S)$	$\pi_{R.a}(R) - \pi_{R.a}(R \cap S)$	No. Consider the following example: R = {(1, 2), (1, 3)}, S={(1, 2), (1,4)} Full credit given to Yes or No answers due to staff error.

$\pi_{R.b}(\sigma_{R.a=1}(R))$	`SELECT R.b`
	`FROM R`
	`WHERE R.a = 1`
		No. Q1 returns a set, Q2 returns a multiset (has duplicates)

$R \bowtie S$	$\sigma_{R.a=S.a}(R \times S)$	No. Q1 is a natural join (on a and b). Q2 is a join only on a

`SELECT * FROM R`
`WHERE R.a < ALL (`
`SELECT a FROM S`
`)`
	`SELECT R.a, R.b`
	`FROM R JOIN S`
	`WHERE R.a < S.a`
		No. Q1 returns R rows less than ALL S.a values.
		Q2 returns R rows less than ANY S.a value.

6 (Extra credit) Wisdom of the Crowds

The following questions are for extra credit. Each answer should be a single integer number.

1. (1.5 points) Guess to within 1 point what your midterm exam score will be when the exams are handed back (not including the extra credit).

2. (1.5 points) Guess to within 1 point what the class average exam score will be when the exams are handed back (not including the extra credit).