Name: \_\_\_\_\_

# Database Systems — CSci 4380 Midterm Exam #2November 1, 2018

@rpi.edu

RCS ID: \_\_\_\_\_

: 1.	ps with start dates at 1		

ouse listed in al	ny other state.			

	(12 points) Find all unconfirmed trips for a house in Troy, New York that were supposed to start on $^{10}/^{31}/^{2018}$ . For each trip, return the house id, house label, email of the owner as well as number of messages sent by the owner for this trip.
(e)	(10 points) Delete all reviews entered by users who are also owners of houses. Such users should have no business reviewing others.

Question 2 (14 points). Write a trigger that activates after a tuple is inserted into the Reviews table. Recall that each review is for a trip and each trip is for a specific house.

The trigger updates the tuple in relation Houses for this house and sets the average rating attribute (Houses.avgrating) to the average of the Reviews.isgoodvalue values for all trips involving this house.

Complete the trigger description given below for simplicity. You are free to use any pl/pgsql type procedural code here.

CREATE FUNCTION avgreview_f () RETURNS trigger AS \$\$ DECLARE	
DECTN	
BEGIN	
RETURN NEW; END;	
\$\$ LANGUAGE plpgsql;	
CREATE TRIGGER avgreview_trigger AFTER INSERT ON reviews	
FOR EACH ROW EXECUTE PROCEDURE avgreview_f();	

Question 3 (a-10 points). Create a view called housestats that returns for each house in the Houses table the following attributes: Houses.id, Houses.label, Houses.state, Houses.city, numtrips (total number

eate view housestats(id	l, label, state, city, nu	umtrips, numdaysrented) as	
		n 3(a) above to find the most purn all the attributes in the view	

Question 4 (12 points). You are given the following table definitions and table contents.

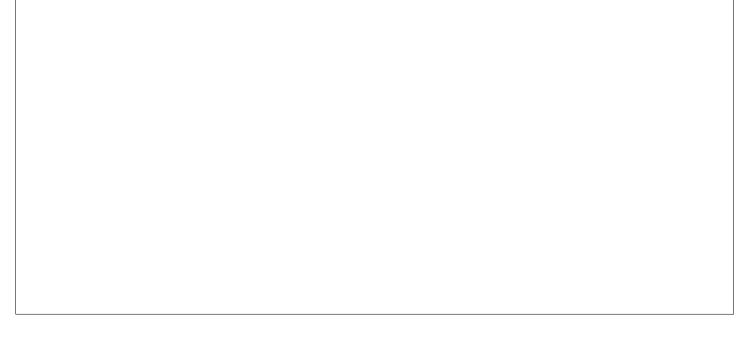
```
CREATE TABLE ROLES(id INT PRIMARY KEY, name VARCHAR(10) NOT NULL, type VARCHAR(10)); CREATE TABLE SHOWS(id INT PRIMARY KEY, title VARCHAR(100) NOT NULL); CREATE TABLE APPEARS(id INT PRIMARY KEY, sid INT, rid INT NOT NULL
```

- , FOREIGN KEY (sid) REFERENCES SHOWS(id) ON DELETE CASCADE ON UPDATE SET NULL
- , FOREIGN KEY (rid) REFERENCES ROLES(id) ON DELETE CASCADE ON UPDATE SET NULL);

Roles			Shows	A	Appea	rs	
id	name	$_{\mathrm{type}}$	$\operatorname{id}$	title	id	$\operatorname{sid}$	$\operatorname{rid}$
1	Rick	Rick	5	Meeseeks and Destroy	1	5	1
2	Morty	Morty	12	A Rickle in Time	2	5	2
3	PickleRick	Rick	24	Pickle Rick	3	24	1
4	Evil Rick	Rick	28	The Ricklantis Mixup	4	24	2
5	Rick Morty	Morty			5	24	3
6	Jaguar				6	24	6
					7	28	1
					8	28	2
					9	28	5

For each operation below, describe which rows from which tables are changed/deleted and why (or why not). Assume each operation operates on the table contents listed above (hence each part is independent).

- (a) DELETE FROM roles WHERE name = 'Jaguar';
- (b) UPDATE shows SET id = 24 WHERE name LIKE '%Rick%';
- (c) UPDATE roles SET id = 8 WHERE id = 5;
- (d) DELETE FROM shows WHERE NOT EXISTS (SELECT \* FROM appears WHERE sid=shows.id);



Use this page for scratch work only. with anyone.	Do not share your solutions or any drafts of your solutions

## Data model to be used in Exam #2

This is a data model based on the E-R problem from Homework#3, storing information about house rentals in a system similar to Airbnb. It is simplified in various ways to make it easier to use in an exam. Please read carefully. The keys of each relation are underlined.

### Users(email, password, name, street, state, city, zip, country, ccno)

Stores information about all the users in the system, renters and customers including credit card number (ccno).

#### Houses(id, label, description, street, state, city, zip, country, price, owneremail, avgrating)

Each house is listed by an owner and we store the email of the owner for each house. The daily price of a house is fixed for simplicity. Average rating is the average rating value for this house.

#### HouseAmenities(houseid, amenity)

This relation lists the various amenities a house has such as 'free wifi', 'garage parking', etc,

#### Trips(id, fromdate, todate, isconfirmed, totalprice, customeremail, houseid)

Each trip has a start and end date, whether it is confirmed or not (true/false value) and a total price for the trip. Each trip is by a user (customer), so his/her email is stored. This is the person paying for the trip. Each trip is for a specific house, so the id of the house is also stored. (The renter can be deduced from the owner of the house).

#### Reviews(tripid, isclean, isgoodvalue, isgoodrenter, reviewtext)

We will only store the reviews of the houses in this database. There can only be one review for each trip, so we will use tripid as the key. Each review is for a specific house and its renter (stored in other relations) and contains 1-5 star values for isclean, isgoodvalue, isgoodrenter as well as a review text.

#### Messages(<u>id</u>, senderemail, receiveremail, messagetext, sentdatetime, prevmid, tripid)

Finally, we store messages from one user (sender) to another user (receiver) together with the text and datetime it was sent. The attribute prevmid can be null if this is an original email or store the id of the message that this message was sent in response to. Each message is for a specific trip.