INF1343, Winter 2012, Quiz 3

_____

write your name above

a) Bob, the owner of Diveshop, wants to find out the total number of rentals for diving masks since the beginning of this year. Assume that the current year is still 2012. Consider that Diveshop has several kinds of diving masks. They are all have their "equipment class" identified as "Mask". Each kind of mask may have been rented out several times and one order may involve rental of several masks. either of the same kind or of different kinds. Bob is interested in the **total**. In other words, if a customer rented out 2 units of "Tri-Vent Mask – Clear" mask and 3 units or "Quad Vision Mask – Red" mask, this order would add 5 to the total. Write the necessary query below. **A sheet with create statements for the Diveshop database is provided for your reference.**

```
select sum(order_item.quantity)
from stocked_item
  join order_item using (item_id)
  join vacation_order using (order_id)
where
  stocked_item.equipment_class="Mask"
  and order_item.rental_or_sale="rental"
  and vacation_order.sale_date>="2011-01-01";
```

Notes:

- You need to join three tables. It has to be those three tables. The order of joins can vary a bit, though, as long as "order_item" is one of the first two tables to be included. (In other words, you cannot join stocked_item to vacation_order, but you can join either to order_item to get started. The important thing is that the field that is introduced in the using clause is present in the table that is currently being joined and in the result of the previous joins.
- You could use "join on …" instead of "join using."
- You have to use "sum" to add up the quantities. Using "count" will give you the total number of orders that included a rental of diving masks, without considering that in some cases multiple masks were rented as a part of the same order.
- If you add any other attributes to your projection, the results you would get in those columns would be meaningless or useless. E.g., if you do "select sum(order_item.quantity), order_item.rental_or_sale" then the value in the second column would either be always "rental" (if you wrote your "where" clause correctly) or it would be be arbitrarily either "rental" or "sale" (if you did not do you "where" clause correctly).
- For date comparison, see the answers to the first quiz.

b) Imagine that Bob told you that the Diveshop database was based on an ER diagram, which was then converted into relational form using the same process that we have discussed in class. Of the nine tables shown on the attached sheet, eight correspond to entities that were present in Bob's *original* ER diagram. One table, however, corresponds to an *associative entity* that Bob created when he broke up M:M relationships. Which table corresponds to the associative entity and which two tables correspond to the two entities that were linked with the M:M relationship that had to be broken up? (Please answer on the other side of the sheet.)

The table corresponding to the associative entity is "site_species". The tables corresponding to the entities that were linked with an M:M relationship are "site" and "species."

Notes:

- “site_species” is the only table that corresponds to a “pure” associative entity. It records no information apart from associating a site and a species. We could replace it with a M:M relationship and not lose any information. The only reason we have this table is because we have no other ways to represent M:M relationships.
- “order_item” is another candidate, but the entity it corresponds to is not purely associative. It does not merely record an association, but also the amount and whether the item was sold or rented.
- None of the other tables really make plausible candidates at all.

INF1343, Winter 2012, Quiz 3 — Create Statements for Some of the Tables in the Diveshop Database

```
create table destination (
  destination_id int(11) not null,
  destination_name varchar(255),
  accommodations varchar(255),
  night_life varchar(255),
  body_of_water varchar(255),
  travel_cost double,
  primary key (destination_id)
);

create table site (
  site_id int(11) not null,
  destination_id int(11),
  site_name varchar(255),
  site_highlight varchar(255),
  site_notes varchar(255),
  distance_from_town_km double,
  depth_m double,
  visibility_m double,
  `current` varchar(255),
  skill_level varchar(255),
  primary key (site_id),
  foreign key (destination_id)
    references destination(destination_id)
);

create table species (
  species_id int(11) not null,
  category varchar(255),
  common_name varchar(255),
  species_name varchar(255),
  length_cm double,
  notes text,
  graphic_file varchar(255),
  primary key (species_id)
);

create table site_species (
  site_id int(11) not null,
  species_id int(11) not null,
  foreign key (site_id)
    references site(site_id),
  foreign key (species_id)
    references species(species_id)
);

create table customer (
  customer_id int(11) not null,
  name varchar(255) not null,
  street varchar(255),
  city varchar(255),
  state_prov varchar(255),
  zip_postal_code varchar(255),
  country varchar(255),
  phone varchar(255),
  first_contact datetime,
  primary key (customer_id)
);
```

```
create table shipment_method (
  shipment_method_id int(11) not null,
  shipment_method_name varchar(255) not null,
  cost decimal(9,2),
  primary key (shipment_method_id)
);

create table vacation_order (
  order_id int(11) not null,
  customer_id int(11),
  destination_id int(11),
  sale_date datetime,
  shipment_method_id int(11),
  shipment_cost decimal(9,2),
  payment_method varchar(255),
  cc_number varchar(255),
  cc_exp_date datetime,
  no_of_people smallint(6),
  depart_date datetime,
  return_date datetime,
  cost decimal(9,2),
  primary key (order_id),
  foreign key (customer_id)
    references customer(customer_id),
  foreign key (destination_id)
    references destination(destination_id),
  foreign key (shipment_method_id)
    references shipment_method(
                shipment_method_id)
);

create table stocked_item (
  item_id int(11) not null,
  description varchar(255),
  equipment_class varchar(255),
  on_hand smallint(6),
  reorder_point smallint(6),
  cost decimal(9,2),
  sale_price decimal(9,2),
  rental_price decimal(9,2),
  primary key (item_id)
);

create table order_item (
  order_id int(11),
  item_id int(11),
  rental_or_sale enum('rental', 'sale'),
  quantity smallint(6),
  line_note varchar(255),
  foreign key (order_id)
    references vacation_order(order_id),
  foreign key (item_id)
    references stocked_item(item_id)
);

-- Some tables are not shown, but you do not
-- need them for this quiz.
```