

INF1343, Winter 2012, Answers to Week 5 Exercises

Part 1. Diveshop

Write queries that would use the “diveshop” database to answer the following questions. Each question needs to be answered with *one* query.

- Where is Amanda Gentry going?

```
select destination.destination_name
from destination
  join vacation_order using(destination_id)
  join customer using (customer_id)
where customer.name="Amanda Gentry";
```

- What items are being shipped to Amanda Gentry? Did she buy or rent them?

```
select stocked_item.description, order_item.rental_or_sale
from stocked_item
  join order_item using(item_id)
  join vacation_order using (order_id)
  join customer using (customer_id)
where customer.name="Amanda Gentry";
```

- What is the most expensive item (by sale price) that is available through Diveshop?

```
select stocked_item.description, stocked_item.sale_price
from stocked_item
order by stocked_item.sale_price desc
limit 1;
```

- What is the most expensive item that has actually been bought? Who bought it?

```
select
  stocked_item.description,
  stocked_item.sale_price,
  customer.name
from stocked_item
  join order_item using(item_id)
  join vacation_order using (order_id)
  join customer using (customer_id)
where
  order_item.rental_or_sale="sale"
order by stocked_item.sale_price desc
limit 1;
```

- How much would it cost to travel to a place where a diver whose skill level is “Beginning” can see a fish called “Blue Angelfish”?

```
select min(destination.travel_cost)
from destination
  join site using (destination_id)
  join site_species using (site_id)
  join species using (species_id)
where
  site.skill_level="Beginning"
  and species.common_name="Blue Angelfish";
```

- What is the name of a customer from San Francisco who ordered a trip that might include a visit to the wrecked ship “Star of Scotland”? (You will need to join five tables. The correct answer is “Barbara

Wright".)

```
select customer.name
from shipwreck
  join site using (site_id)
  join destination using (destination_id)
  join vacation_order using (destination_id)
  join customer using (customer_id)
where shipwreck.ship_name="Star of Scotland"
and customer.city="San Francisco";
```

When trying to answer those questions, you might find it handy to draw an ER diagram for the database first.

Part 2. Querying M's Database

Write queries that would answer each of the following questions using the database you built for M last week:

- What is the total value of gadgets currently in possession of agent 007?

This query will depend on how you implemented this database, but if your implementation followed the ER diagram from last week, you would be doing two joins: agent to gadget_loan to gadget_unit. Your where condition would specify that agent's code must be 007 and your projection would sum up the values.

- How many units of "X-ray glasses" have been lost by agent 007?

If your implementation followed the provided ER diagram then you would need the same join as for the previous question, and then also add gadget_type, specifying that you are interested in "X-ray glasses." Your where clause would also add a condition that gadget_loan's FK to loss_report is not null. (There is no need to actually join loss_report, since you do not need any information from it. You just need to select those cases where a loss report exists.)

- How many gadgets were lost in all operations involving enemy "SPECTRE"?

Four joins: gadget_unit to gadget_loan to loss_report to operation to enemy. The where clause would specify that the name of enemy must be "SPECTRE" and your projection would just count the gadgets.

If you did not complete the implementation of M's database last week, please do so now then return to this week's exercises.

Part 3.

Please answer the following questions.

- Would agent's code name (e.g., "007") be a good attribute to use as the primary key for the "agent" table in M's database? Explain.

*No. The fact that the code names like "007" are used to refer to the agents **outside** of the database makes it them "meaningful." Since they are meaningful, a need to change them may arise. The fact that they are numeric and appear to be arbitrary does not change this. It's just like with Social Insurance Numbers.*

- What is the difference between using "JOIN ON ...", "JOIN USING (...)" and "NATURAL JOIN"? When used JOIN ON and JOIN USING (...)

JOIN ON is the general form of join. You can use this for any kind of join. JOIN USING is a shortcut that we can use for an equi-join when the names of the joined attributes match. There is no downside to using it in this situation, but there are many situations where we cannot use it, either because the join we want to do is not an equi-join or because the names of the attributes involved in the equi-join do not match. NATURAL JOIN basically asks the database to figure out which fields should be matched in an equi-join. This saves you from doing a little bit of thinking and typing but makes the result unpredictable. Don't use it.

- Can any ER model be implemented as a relational database? Can any relational database be modeled using an ER diagram? If the answer to either question is “no”, what would be an example?

*The answer to both questions is “no.” ER diagrams allow for 1:M relationships that are mandatory on the “M” side. This specific constraint cannot be expressed in a relational form. So, ER diagrams with such relationships cannot be converted into relational form precisely. Apart from that, practically speaking, any ER diagram can be converted to a relational form. So, the answer to the first question is, technically, “no.” The answer to the second question is a strong “no.” The relational model is much much more flexible than the ER model. We can easily implement databases for which there would be no meaningful ER representation. Take a simple database with a single table that has two columns: one storing names of people and another one storing titles of books. There is no meaningful ER diagram that would represent. (In most cases, though, a relational organization that cannot be expressed in an ER diagram is not a **good** relational organization.)*

- When we do a relational join, does it matter if the columns we use in the join condition form PK-FK pairs?

*Generally speaking, no. We **often** do joins on attributes forming PK-FK pairs. In fact, the whole point of having PK-FK pairs is to use them in joins later. However, we sometimes do joins that use other attributes. One of the examples that we used in class we wanted to get pairs of animals have the same species. This join was not based on attributes forming a PK-FK pair.*