Week 2

Theory 1

Relational algebra operators:

- Set operations:
    - Union: $R \cup S$, elms in R or S
    - Intersection: $R \cap S$, elms in R and S
    - Difference: $R - S$, elms in R but not in S
    - Need schemas with identical sets of attributes and domains. Order should be the same as well.
- Projection:
    - Removes some columns
    - Denoted $\pi_{a,b,c}(R)$
        - Gives columns a, b, c
- Selection:
    - Removes some rows
    - Gives subset of entries
    - Denoted $\sigma_C(R)$
        - Where C is some condition, e.g. $length \geq 100$
- Cartesian product
    - Aka. Cross product, aka product
    - Denoted $R \times S$
    - Set of pairs formed by choosing first in R second in S
    - Schema is the union of the schemas of R and S
        - If attribute names in common, we invent new names $R.A$, $S.A$
- Joins
    - Natural join:
        - Denoted $R \bowtie S$
        - Pair tuples that agree in whatever attributes common in their schemas
    - Theta join:
        - Denoted $R \bowtie_C S$
            - Where C is some condition
        - A join on some condition e.g. $R \bowtie_{A<D} S$
- Renaming
    - Renames both the relation and the attributes
    - Denoted $\rho_{S(a,b,c)}(R)$
        - Renames relation R to S with attributes a, b, c
- Linear notation:

R(t,y,l,i,s,p) := $\sigma_{length \geq 100}$(Movies)
S(t,y,l,i,s,p) := $\sigma_{studioName='Fox'}$(Movies)
T(t,y,l,i,s,p) := R $\cap$ S
Answer(title, year) := $\pi_{t,y}$(T)

Bags:

- “multiset”: allow element to be present more than once
- How implemented in DBMS
- Some operations more efficient on bags
  - Union: just copy (no duplicate elimination)
  - Projection no duplicate elimination
- R, S are bags with t appearing n, m times respectively:
  - $R \cup S$: t appears $n + m$ times
  - $R \cup S$: t appears $\min(n, m)$ times
  - $R - S$: t appears $\max(0, n - m)$ times

SQL:

- Pattern matching: $s\ LIKE\ p$
  - Where p is some pattern
  - _ means a char
  - % means 0 or more chars
- Values could be $NULL$
  - Using operators means getting truth value $UNKNOWN$

x	y	x AND y	x OR y	NOT x
TRUE	TRUE	TRUE	TRUE	FALSE
TRUE	UNKNOWN	UNKNOWN	TRUE	FALSE
TRUE	FALSE	FALSE	TRUE	FALSE
UNKNOWN	TRUE	UNKNOWN	TRUE	UNKNOWN
UNKNOWN	UNKNOWN	UNKNOWN	UNKNOWN	UNKNOWN
UNKNOWN	FALSE	FALSE	UNKNOWN	UNKNOWN
FALSE	TRUE	FALSE	TRUE	TRUE
FALSE	UNKNOWN	FALSE	UNKNOWN	TRUE
FALSE	FALSE	FALSE	FALSE	TRUE

- Order output by using ORDER BY <list of attr>

Exercises for 3rd exercise session
4.1 (page 49-51)
4.2 (page 51). Draw the trees for at least three expressions from exercise 4, but you don't need to draw them all.

4.5, 4.6, 4.7 (page 53-54)
4.8 (page 54). One expression is enough
4.9 (page 54)

1.2 (page 207), 1.2 (page 250-251), 1.3 (page 252), 2.1 (page 261)

Chapter 2

4.1

Exercise 2.4.1: This exercise builds upon the products schema of Exercise 2.3.1. Recall that the database schema consists of four relations, whose schemas are:

```
Product(maker, model, type)
PC(model, speed, ram, hd, price)
Laptop(model, speed, ram, hd, screen, price)
Printer(model, color, type, price)
```

Some sample data for the relation Product is shown in Fig. 2.20. Sample data for the other three relations is shown in Fig. 2.21. Manufacturers and model numbers have been "sanitized," but the data is typical of products on sale at the beginning of 2007.

Write expressions of relational algebra to answer the following queries. You may use the linear notation of Section 2.4.13 if you wish. For the data of Figs. 2.20 and 2.21, show the result of your query. However, your answer should work for arbitrary data, not just the data of these figures.

a) What PC models have a speed of at least 3.00?

b) Which manufacturers make laptops with a hard disk of at least 100GB?

c) Find the model number and price of all products (of any type) made by manufacturer B.

d) Find the model numbers of all color laser printers.

e) Find those manufacturers that sell Laptops, but not PC's.

! f) Find those hard-disk sizes that occur in two or more PC's.

! g) Find those pairs of PC models that have both the same speed and RAM. A pair should be listed only once; e.g., list (i, j) but not (j, i).

!! h) Find those manufacturers of at least two different computers (PC's or laptops) with speeds of at least 2.80.

!! i) Find the manufacturer(s) of the computer (PC or laptop) with the highest available speed.

!! j) Find the manufacturers of PC's with at least three different speeds.

!! k) Find the manufacturers who sell exactly three different models of PC.

a. $\pi_{model}\left(\sigma_{speed\geq 3.00}(PC)\right)$

b. $\pi_{maker}\left((Product) \bowtie \left(\sigma_{hd\leq 100}(Laptop)\right)\right)$

c. $\pi_{model,price}\big((\sigma_{maker=B}(Product) \bowtie PC) \cup (\sigma_{maker=B}(Product) \bowtie Laptop) \cup (\sigma_{maker=B}(Printer) \bowtie PC)\big)$

d. $\pi_{model}\left(\sigma_{type=laser\ AND\ color=True}(Printer)\right)$

e. $\pi_{maker}\left(\sigma_{type=laptop}(Product)\right) - \pi_{maker}\left(\sigma_{type=pc}(Product)\right)$

f. $\pi_{hd}\big(\rho_{PC2(m,s,r,h,p)}(PC) \bowtie_{pc.hd=pc2.hd\ AND\ pc.model<>pc2.model} PC\big)$

g. $\pi_{pc1.maker,pc2.maker}\big(PC1 \bowtie_{PC1.speed=pc2.speed\ AND\ pc1.RAM=pc2.RAM\ AND\ pc1.model<pc2.model} PC2\big)$

h.

$$R1 \coloneqq \left(\sigma_{speed\geq 2.8}(PC) \cup \sigma_{speed\geq 2.8}(Laptop)\right) \bowtie Product$$

$$R2 \coloneqq \rho(R1)$$

$$\pi_{maker}(R1 \bowtie_{R1.maker=R2.maker\ AND\ R1.model<R2.model} R2)$$

i.

$$R1 \coloneqq \pi_{model,speed}(PC) \cup \pi_{model,speed}(Laptop)$$

$$R2 \coloneqq \rho_{R2(model2,speed2)}(R1)$$

Finding all speeds which is not the largest:

$$R3 \coloneqq \pi_{model,speed}(R1 \bowtie_{speed<speed2} R2)$$

$$R4 \coloneqq R1 - R3$$

$$R5 \coloneqq \pi_{maker}(R4 \bowtie Product)$$

j. …

$$R1 \coloneqq \pi_{maker,speed}(PC \bowtie Product)$$

$$R2 \coloneqq \rho_{R2(maker2,speed2)}(R1)$$

$$R3 \coloneqq \rho_{R3(maker3,speed3)}(R1)$$

$$R4 \coloneqq R1 \bowtie_{speed<>speed2\ AND\ maker=maker2} R2$$

$$R5 = \pi_{maker}\left(R4 \bowtie_{maker=maker3\ AND\ speed<>speed3\ AND\ speed2<>speed3} R3\right)$$

k. …

$$R1 \coloneqq \pi_{maker,model}\left(\sigma_{type=pc}(Product)\right)$$

$$R2 \coloneqq \rho_{R2(maker2,model2)}R1$$

$$R3 \coloneqq \rho_{R3(maker3,model3)}R1$$

$$R4 \coloneqq \rho_{R2(maker4,model4)}R1$$

$$R5 \coloneqq R1 \bowtie_{maker=maker2\ AND\ model<>model2} R2$$

$$R6 \coloneqq R5 \bowtie_{maker=maker3\ AND\ model<>model3\ AND\ model2<>model3} R3$$

$$R7 \coloneqq R4 \bowtie_{maker=maker4\ AND\ (model4=model\ OR\ model4=model2\ OR\ model4=model3)} R6$$

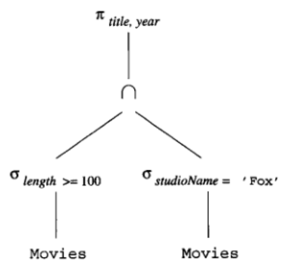
$$R8 \coloneqq \pi_{maker}(R7)$$

4.2

Exercise 2.4.2: Draw expression trees for each of your expressions of Exercise 2.4.1.

Three trees are enough. Made on blackboard.

Trees like this, basically just splitting up the statement.



4.5

Exercise 2.4.5: What is the difference between the natural join $R \bowtie S$ and the theta-join $R \bowtie_C S$ where the condition C is that $R.A = S.A$ for each attribute A appearing in the schemas of both R and S?

Difference is: theta join columns would appear twice, not in natural. Merging vs not merging columns.

4.6

! **Exercise 2.4.6:** An operator on relations is said to be *monotone* if whenever we add a tuple to one of its arguments, the result contains all the tuples that it contained before adding the tuple, plus perhaps more tuples. Which of the operators described in this section are monotone? For each, either explain why it is monotone or give an example showing it is not.

$$\pi, \sigma, \bowtie, \cup, \cap$$

This is not:

$$-$$

4.7

! **Exercise 2.4.7:** Suppose relations R and S have n tuples and m tuples, respectively. Give the minimum and maximum numbers of tuples that the results of the following expressions can have.

a) $R \cup S$.

b) $R \bowtie S$.

c) $\sigma_C(R) \times S$, for some condition C.

d) $\pi_L(R) - S$, for some list of attributes L.

a. $min = \max(m, n)\,, max = n + m$

Min if all tuples in one appears in other

Max if sets disjoint.

b. $min = 0, max = m \cdot n$

Max if all matches all.

c. $min = 0, max = m \cdot n$
d. $min = \max(n - m, 0)\,, max = n$

4.8

! Exercise 2.4.8: The *semijoin* of relations R and S, written $R \ltimes S$, is the set of tuples t in R such that there is at least one tuple in S that agrees with t in all attributes that R and S have in common. Give three different expressions of relational algebra that are equivalent to $R \ltimes S$.

One expression is enough

$$\pi_R(R \bowtie S)$$

4.9

! Exercise 2.4.9: The *antisemijoin* $R \,\overline{\ltimes}\, S$ is the set of tuples t in R that do *not* agree with any tuple of S in the attributes common to R and S. Give an expression of relational algebra equivalent to $R \,\overline{\ltimes}\, S$.

$$R - \left(\pi_R(R \bowtie S)\right)$$

Chapter 5

1.2

!! Exercise 5.1.5: The following algebraic laws hold for sets but not for bags. Explain why they hold for sets and give counterexamples to show that they do not hold for bags.

a) $(R \cap S) - T = R \cap (S - T)$.

b) The distributive law of intersection over union:

$$R \cap (S \cup T) = (R \cap S) \cup (R \cap T)$$

c) $\sigma_{C \text{ OR } D}(R) = \sigma_C(R) \cup \sigma_D(R)$. Here, C and D are arbitrary conditions about the tuples of R.

a. …

For sets:

- Taking intersection subtracting T, same as subtracting from S and taking the intersection
- Does not matter if we take intersection and subtract or other order.

For bags:

- Lhs: something may be apparent more than once in the intersection and subtracted once, meaning seen once totally.

- Rhs: will not have this elm at all due to subtracting elm from S, meaning will not be a part of the intersection with R.

Numbers represent # times some tuple appears in the different relations. $R = 2, S = 3, T = 1$

$$(R \cap S) - T = R \cap (S - T)$$

$$(2 \cap 3) - 1 = 2 \cap (3 - 1)$$

$$2 - 1 = 2 \cap 2$$

$$1 = 2$$

b. …

For sets:

- Intersection with the union same as union of the intersections

For bags:

- If something in all relations, due to allowing duplicates it would appear once on lhs, and twice on rhs.

Numbers represent # times some tuple appears in the different relations. $R = 5, S = 4, T = 4$

$$R \cap (S \cup T) = (R \cap S) \cup (R \cap T)$$

$$5 \cap (4 \cup 4) = (5 \cap 4) \cup (5 \cap 4)$$

$$5 \cap 8 = 4 \cup 4$$

$$5 = 8$$

c. …

For sets:

- Just one of the conditions should hold, then in the result set. This is the same as the union due to duplicates being eliminated

For bags:

- Duplicates not being eliminated, meaning that if both conditions hold, some tuple would be once on lhs and twice on rhs.

R(age, height):

- (35, 170)
- (30, 180)

$$\sigma_{age<40} R \ OR \ \sigma_{height<190} \ R = \sigma_{age<40} R \cup \sigma_{height<190} \ R$$

$$|2| = |4|$$

Chapter 6

1.2

Exercise 6.1.2: Write the following queries, based on our running movie database example

```
Movies(title, year, length, genre, studioName, producerC#)
StarsIn(movieTitle, movieYear, starName)
MovieStar(name, address, gender, birthdate)
MovieExec(name, address, cert#, netWorth)
Studio(name, address, presC#)
```

in SQL.

a) Find the address of MGM studios.

b) Find Sandra Bullock's birthdate.

c) Find all the stars that appeared either in a movie made in 1980 or a movie with "Love" in the title.

d) Find all executives worth at least $10,000,000.

e) Find all the stars who either are male or live in Malibu (have string `Malibu` as a part of their address).

```sql
SELECT address
FROM Studio
WHERE name = 'MGM';

SELECT birthdate
FROM MovieStar
WHERE name = 'Sandra Bullock';

SELECT starName
FROM Movies, StarsIn
WHERE year = 1980 OR title LIKE '%Love%';

SELECT name
FROM MovieExec
WHERE netWorth >= 10000000;

SELECT name
FROM MovieStar
WHERE gender = 'male' OR address = 'Malibu';
```

1.3

Exercise 6.1.5: Let a and b be integer-valued attributes that may be `NULL` in some tuples. For each of the following conditions (as may appear in a `WHERE` clause), describe exactly the set of (a, b) tuples that satisfy the condition, including the case where a and/or b is `NULL`.

a) `a = 10 OR b = 20`

b) `a = 10 AND b = 20`

c) `a < 10 OR a >= 10`

! d) `a = b`

! e) `a <= b`

a. …

$a = 10$	OR	$b = 20$
$True$		$True$
$True$		$False$
$False$		$True$
$True$		$Null$

b. …

$a = 10$	AND	$b = 20$
$True$		$True$

c. …

$a < 10$	OR	$a \geq 10$
$True$		$False$
$False$		$True$

d. …

a	$=$	b
$NOT\ NULL$	$when\ a = b$	$NOT\ NULL$

e. …

a	$\leq$	b
$NOT\ NULL$	$when\ a \leq b$	$NOT\ NULL$

2.1

Exercise 6.2.1: Using the database schema of our running movie example

```
Movies(title, year, length, genre, studioName, producerC#)
StarsIn(movieTitle, movieYear, starName)
MovieStar(name, address, gender, birthdate)
MovieExec(name, address, cert#, netWorth)
Studio(name, address, presC#)
```

write the following queries in SQL.

a) Who were the male stars in *Titanic*?

b) Which stars appeared in movies produced by MGM in 1995?

c) Who is the president of MGM studios?

! d) Which movies are longer than *Gone With the Wind*?

! e) Which executives are worth more than Merv Griffin?

```
SELECT starname
FROM starsin, moviestar
WHERE movietitle = 'titanic' AND gender = 'male' AND starname = name;
```

```
SELECT starName
FROM StarsIn, Movies
WHERE year = 1995 AND studioName = 'MGM' AND title = movieTitle;
```

```
SELECT MovieExec.name
FROM Studio, MovieExec
WHERE presidentCert = cert AND Studio.name = 'MGM';
```

```
SELECT title
FROM Movies
WHERE length > (SELECT length
                FROM Movies
                WHERE title = 'Gone With the Wind');
```

```
SELECT name
FROM MovieExec
WHERE netWorth > (SELECT netWorth
                  FROM MovieExec
                  WHERE name = 'Merv Griffin');
```