1.

title
ACADEMY DINOSAUR
AFFAIR PREJUDICE
AGENT TRUMAN
AIRPLANE SIERRA
ALABAMA DEVIL
ALADDIN CALENDAR
ALAMO VIDEOTAPE
ALASKA PHANTOM
ALIEN CENTER
ALLEY EVOLUTION

This query returned 563 rows

2.

film_id	title	type	last_update
19	AMADEUS HOLY	actor	2006-02-15 05:05:03
19	AMADEUS HOLY	category	2006-02-15 05:07:09
19	AMADEUS HOLY	inventory	2006-02-15 05:09:17
21	AMERICAN CIRCUS	actor	2006-02-15 05:05:03
21	AMERICAN CIRCUS	category	2006-02-15 05:07:09
21	AMERICAN CIRCUS	inventory	2006-02-15 05:09:17
29	ANTITRUST TOMATOES	actor	2006-02-15 05:05:03
29	ANTITRUST TOMATOES	category	2006-02-15 05:07:09
29	ANTITRUST TOMATOES	inventory	2006-02-15 05:09:17
38	ARK RIDGEMONT	actor	2006-02-15 05:05:03

This query returned 189 rows

3.

film_id	title	rating	length	type
1	ACADEMY DINOSAUR	PG	86	feature
2	ACE GOLDFINGER	G	48	short
3	ADAPTATION HOLES	NC-17	50	short
4	AFFAIR PREJUDICE	G	117	feature
5	AFRICAN EGG	G	130	feature
6	AGENT TRUMAN	PG	169	feature
7	AIRPLANE SIERRA	PG-13	62	featurette
8	AIRPORT POLLOCK	R	54	featurette
9	ALABAMA DEVIL	PG-13	114	feature
10	ALADDIN CALENDAR	NC-17	63	featurette

This query returned 1000 rows

4.

film_id	title	rating	length	rank
237	DIVORCE SHINING	G	47	1
247	DOWNHILL ENOUGH	G	47	1
2	ACE GOLDFINGER	G	48	3
575	MIDSUMMER GROUNDHOG	G	48	3
430	HOOK CHARIOTS	G	49	5
83	BLUES INSTINCT	G	50	6
292	EXCITEMENT EVE	G	51	7
402	HARPER DYING	G	52	8
542	LUST LOCK	G	52	8
794	SIDE ARK	G	52	8

This query returned 1000 rows

5.

film_id	title	rating	length
237	DIVORCE SHINING	G	47
247	DOWNHILL ENOUGH	G	47
2	ACE GOLDFINGER	G	48
575	MIDSUMMER GROUNDHOG	G	48
430	HOOK CHARIOTS	G	49
83	BLUES INSTINCT	G	50
292	EXCITEMENT EVE	G	51
469	IRON MOON	PG	46
784	SHANGHAI TYCOON	PG	47
869	SUSPECTS QUILLS	PG	47

This query returned 45 rows

6.

film_id	actors
1	10
2	4
3	5
4	5
5	5
6	7
7	5
8	4
9	9
10	8

This query returned 1000 rows, there are 3 movies with 0 actors

7. Assuming I have the following relations (may be subject to change but query will keep the same concept):

- recipes (recipe_id, title, description, creator_id, create_date, last_update)
- reviews (review_id, recipe_id, user_id, rating, comment, create_date, last_update)

The query will use a LEFT JOIN to include all recipes, even those without reviews. The query will then use COUNT, GROUP BY and ORDER BY to display to the user the most-reviewed recipes.

8. Using the same tables from above, my query involving a CASE statement will categorize recipes into different levels (Excellent, Good, Average, Poor, Very Poor) based on the average rating.

9. Assuming I have the following relations:

- recipes (recipe_id, title, description, creator_id, create_date, last_update)
- ingredients (ingredient_id, name)
- recipe_ingredients (recipe_id, ingredient_id, quantity)

The query will use a window function to calculate the average number of ingredients per recipe to give insight into the average complexity of each recipe.