Joins Exercise 6-1

Using sakila database – Checkpoint Lab

1. Get a list of all film titles and their inventory number – even those we don't have.
   - You might like to limit the output to 200 to check you have everything looking like this

title	inventory_id
ACADEMY DINOSAUR	1
ACADEMY DINOSAUR	2
ACADEMY DINOSAUR	3
ACADEMY DINOSAUR	4
ACADEMY DINOSAUR	5
ACADEMY DINOSAUR	6
ACADEMY DINOSAUR	7
ACADEMY DINOSAUR	8

2. Which films do we ***not*** have in stock?

title
ALICE FANTASIA
APOLLO TEEN
ARGONAUTS TOWN
ARK RIDGEMONT
ARSENIC INDEPENDENCE
BOONDOCK BALLROOM
BUTCH PANTHER
CATCH AMISTAD

   - There are 42 records beginning with:

3. List the number of films in which each actor has featured (sort the output in descending order of the number of films)

first_name	last_name	film_count
GINA	DEGENERES	42
WALTER	TORN	41
MARY	KEITEL	40
MATTHEW	CARREY	39
SANDRA	KILMER	37
SCARLETT	DAMON	36
VAL	BOLGER	35

Techniques from this week

4. The store uses a formula to calculate the return-on-investment (or **ROI**) which is (rental_rate / replacement_cost * 100). List the films, rental replacement cost and **ROI** which have an **ROI** more than 10. Order by ROI. ***Only have the formula once in the query***

title	rental_rate	replacement_cost	roi
ARIZONA BANG	2.99	28.99	10.313901
MONSTER SPARTACUS	2.99	28.99	10.313901
ICE CROSSING	2.99	28.99	10.313901
FOREVER CANDIDATE	2.99	28.99	10.313901
SEABISCUIT PUNK	2.99	28.99	10.313901
RIDER CADDYSHACK	2.99	28.99	10.313901
ZOOLANDER FICTION	2.99	28.99	10.313901

5. List the *maximum*, *minimum* and average film replacement cost using subselects in the select clause only (do not use a **FROM** clause in the *main* query) – yes this is silly.

Max	Min	avg
29.99	9.99	19.984000

Student Database on SQLite *Techniques from earlier this week*

Not trivial, you'll have to work on these.

6. List the students as pairs who come from the same sized high school. Order by school size.
   - Only list one pair of each student e.g. if you have Alice and Bob in a record don't also list Bob and Alice (unless they are different students – we have two different AMY's).
   - Work through this in stages – removing redundant pairs is the last step. You might like to display more information while developing the query (e.g. sid)

Student1	Student2	School Size
Gary	Helen	800
Amy	Doris	1000
Amy	Amy	1000
Doris	Amy	1000
Bob	Jay	1500
Craig	Edward	2000

7. List each student that has made an application and the number of ITP's they have applied to.

sName	ITP_Count
Amy	3
Bob	1
Craig	2
Fay	1
Helen	2
Irene	2
Jay	2

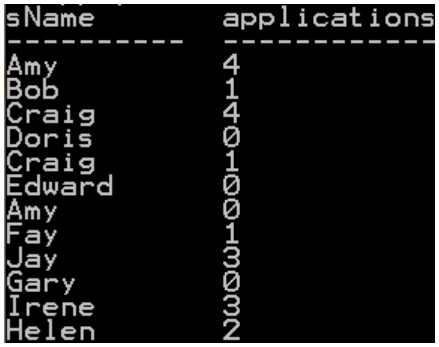
   - This is a simple inner join but I did use something introduced in passing today to get this.

Subqueries and Outer Joins

8. Which students have not applied anywhere?

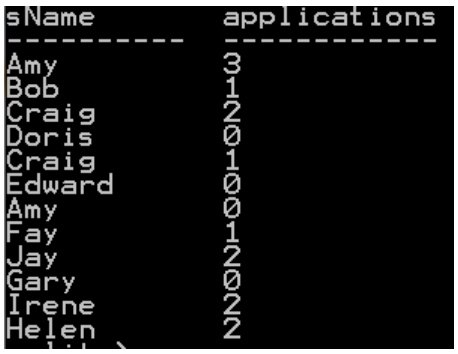
sName
Doris
Edward
Gary
Amy

9. List a count of the number of applications made by each student



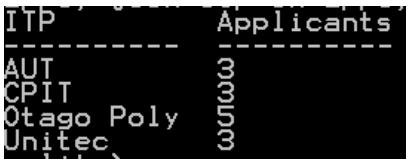
sName	applications
Amy	4
Bob	1
Craig	4
Doris	0
Craig	1
Edward	0
Amy	0
Fay	1
Jay	3
Gary	0
Irene	3
Helen	2

10. List the number of institutions that each student has applied to:



sName	applications
Amy	3
Bob	1
Craig	2
Doris	0
Craig	1
Edward	0
Amy	0
Fay	1
Jay	2
Gary	0
Irene	2
Helen	2

11. How many students have applied to each institution?



ITP	Applicants
AUT	3
CPIT	3
Otago Poly	5
Unitec	3