

1)

Names of the all the artists who have albums to their name

Query: **select DISTINCT artists.Name as "artists who have albums" from albums,artists where albums.ArtistId=artists.ArtistId;**

Names of the all the artists who do not have albums to their name

Query: **SELECT artists.Name as "artists who do not have albums" FROM artists WHERE (ArtistId NOT IN (SELECT ArtistId FROM albums));**

for each first alphabet of the artist name, the count of artists per alphabet.

Query: **select substring(Name,1,1) as "first alphabet of the artist name", count(*) as "count of artists per alphabet" from (select artists.Name from albums,artists where albums.ArtistId=artists.ArtistId UNION ALL SELECT artists.Name FROM artists WHERE (ArtistId NOT IN (SELECT ArtistId FROM albums))) GROUP by substring(Name,1,1);**

first alphabet of the artist name	count of artists per alphabet
A	32
B	27
C	25
D	19
E	14
F	15
G	15
H	6
I	24
J	19
K	5
L	31
M	33
N	8
O	18
P	15
Q	3
R	16
S	32
T	29
U	11
V	14
W	2
X	1
Y	3
Z	1

2)

Write a query that "emulates" a FULL OUTER JOIN on the student and instructor tables listing the instructor's name, the course they teach, the student's name, and the course they take using just LEFT JOIN.

Query: **SELECT instructor.name as "instructor_name",instructor.course_id as "course they teach" ,student.name as "student_name", student.course_id as " course they take" FROM instructor LEFT JOIN student ON instructor.course_id=student.course_id**

union

SELECT instructor.name as "instructor_name",instructor.course_id as "course they teach" ,student.name as "student_name", student.course_id as " course they take" FROM student LEFT JOIN instructor ON instructor.course_id=student.course_id;

instructor_name	course they teach	student_name	course they take
		Jack	CS800
		Jason	CS450
Aaron	CS700		
Amy	CS1000	Jones	CS1000
Anne	CS400		

3)

Report the count of tracks per play length classification.

Query: **select play_length,count(*) as "count of tracks" from (select**

case

when Milliseconds<(60000) then 'short'

when Milliseconds between (60000) and (300000) then 'medium'

when Milliseconds>(300000) then 'long'

end as 'play_length' from tracks) group by play_length;

play_length	count of tracks
Long	1069
medium	2407
short	27

4)

For all entries in table X, generate all possible combinations of size 3 for both the numeric IDs (id num) and corresponding string IDs (id str).

Query: **SELECT p.id_num||','||q.id_num||','||r.id_num as "ID_num",
p.id_str||','||q.id_str||','||r.id_str AS "ID_str"**

from X as p,X as q,X as r where p.id_num<q.id_num and q.id_num<r.id_num ;

ID_num	ID_str
1,2,3	A,B,C
1,2,4	A,B,D
1,2,5	A,B,E
1,3,4	A,C,D
1,3,5	A,C,E
1,4,5	A,D,E
2,3,4	B,C,D
2,3,5	B,C,E
2,4,5	B,D,E
3,4,5	C,D,E