

[Return to Classroom](#)[DISCUSS ON STUDENT HUB](#)

Investigate a Relational Database

REVIEW

CODE REVIEW 5

HISTORY

▼ Queries.project1.txt 5

```
1 Question 1 for slide 1
2 -----
3 Query 1.
4 Create a query that lists each movie, the film category it is classified in, and the number of times it has
5
6 SELECT DISTINCT(f.title) film_title,
7         c.name category_name,
8         COUNT(r.rental_date) OVER (PARTITION BY f.title ORDER BY f.title) AS rental_count
9 FROM film f
10 JOIN film_category fc
11 ON f.film_id = fc.film_id
```

AWESOME

Way to go! Your query ran without any errors and returned the correct results.

```
12 JOIN category c
13 ON fc.category_id = c.category_id
14 JOIN inventory i
15 ON f.film_id = i.film_id
16 JOIN rental r
17 ON i.inventory_id = r.inventory_id
18 WHERE c.name = 'Music'
19        OR c.name = 'Family'
20        OR c.name = 'Children'
```

SUGGESTION

Another alternative to using multiple OR clauses is by doing this:

```
WHERE c.name IN ('Animation', 'Classics', 'Children', 'Comedy', 'Family', 'Music')
```

```
21 OR c.name = 'Classics'
22 OR c.name = 'Comedy'
23 OR c.name = 'Animation'
24 ORDER BY c.name, f.title;
25
26 Question 2 for slide 2
27 -----
28 Query 2.
29 Can you provide a table with the movie titles and divide them into 4 levels (first_quarter, second_quarter,
30
31 SELECT f.title, c.name, f.rental_duration,
32        NTILE(4) OVER(ORDER BY f.rental_duration) AS standard_quartile
33 FROM category AS c
34
```

AWESOME

Way to go! Your query ran without any errors and returned the correct results.

```
35 JOIN film_category AS fc
36 ON c.category_id = fc.category_id
37 AND c.name IN ('Animation', 'Children', 'Classics', 'Comedy', 'Family', 'Music')
38 JOIN film AS f
39 ON f.film_id = fc.film_id
40 ORDER BY 4;
41
```

```

42 Question 3 for slide 3
43 -----
44
45 Query 3.
46 provide a table with the family-friendly film category, each of the quartiles, and the corresponding count
47
48 .Category
49 .Rental length category
50 .Count
51
52 SELECT
53     category,
54     percentile_rental_duration,
55     count(*)
56 FROM(
57 SELECT
58     c.name AS category,
59     f.rental_duration AS rental_duration,
60     NTILE(4) OVER ( ORDER BY f.rental_duration) AS percentile_rental_duration

```

AWESOME

Way to go! Your query ran without any errors and returned the correct results.

```

61 FROM film f
62 JOIN film_category fc
63 ON f.film_id=fc.film_id
64 JOIN category c
65 ON fc.category_id=c.category_id
66 WHERE c.name IN ('Animation','Children','Classics','Comedy','Family','Music')
67 ) sub
68 GROUP BY 1,2
69 ORDER BY 1,2;
70

```

```

71 Question 4 for slide 4
72 -----
73
74 Query 4.
75 query that returns the store ID for the store, the year and month and the number of rental orders each store
76
77 SELECT DATE_PART('month', r.rental_date) Rental_month,
78         DATE_PART('year', r.rental_date) Rental_year,

```

AWESOME

Way to go! Your query ran without any errors and returned the correct results.

```

79     s.store_id Store_id,
80     count(*) Rentals_count
81 FROM rental r
82 JOIN staff sf
83 ON r.staff_id = sf.staff_id
84 JOIN store s
85 ON s.store_id = sf.store_id
86 GROUP BY 1,2,3
87 ORDER BY 4 desc;

```