

```

1  -- Kevin Hance
2  -- HW8
3
4  -- READING ASSIGNMENT
5
6  -- Reading assignment question 2
7  WITH t AS (SELECT * FROM actor a WHERE a.actor_id > 20)
8      SELECT * FROM t ORDER BY t.first_name DESC;
9
10 -- Reading assignment question 3
11 DROP TABLE IF EXISTS table_name;
12
13 CREATE TABLE table_name(
14     column1 INT,
15     column2 INT,
16     column3 INT,
17     PRIMARY KEY (column1)
18 );
19
20 INSERT INTO table_name VALUES (1, 2, 3);
21 INSERT INTO table_name VALUES (4, 5, 6);
22 INSERT INTO table_name VALUES (7, 8, 9);
23
24 UPDATE table_name
25 SET
26     column2 = 6,
27     column3 = 10
28 WHERE
29     column3 = 6;
30
31 SELECT * FROM table_name;
32
33 UPDATE table_name
34 SET
35     column2 = 10,
36     column3 = 15
37 WHERE
38     column3 > 5;
39
40 SELECT * FROM table_name;
41
42 DELETE FROM table_name WHERE column1 = 1;
43
44 SELECT * FROM table_name;
45
46 DELETE FROM table_name WHERE column2 > 0;
47
48 SELECT * FROM table_name;
49
50 -- Reading assignment question 4
51
52 DROP TABLE IF EXISTS people;
53 DROP TABLE IF EXISTS people_location;
54
55 CREATE TABLE people(
56     user_identification INT UNSIGNED,
57     first_name VARCHAR(30),
58     last_name VARCHAR(30),
59     age INT UNSIGNED,
60     PRIMARY KEY (user_identification)
61 );
62
63 INSERT INTO people VALUES (0, "Kevin", "Hance", 21);
64 INSERT INTO people VALUES (1, "Jane", "Hance", 59);
65 INSERT INTO people VALUES (2, "Barry", "Hance", 58);
66
67 CREATE TABLE people_location(
68     user_identification INT UNSIGNED,
69     location_city VARCHAR(30),

```

```

70     PRIMARY KEY (user_identification)
71 );
72
73 INSERT INTO people_location VALUES (0, "Spokane");
74 INSERT INTO people_location VALUES (1, "Woodinville");
75 INSERT INTO people_location VALUES (2, "Woodinville");
76
77 SELECT * FROM people;
78 SELECT * FROM people_location;
79
80 SELECT *
81 FROM people RIGHT JOIN people_location
82 ON people.user_identification = people_location.user_identification;
83
84 -- TECHNICAL WORK
85
86 -- Question 1
87 SELECT a.first_name, a.last_name, count(f.film_id)
88 FROM actor a JOIN film f JOIN film_actor fa
89 ON a.actor_id = fa.actor_id AND f.film_id = fa.film_id
90 GROUP BY a.actor_id
91 ORDER BY count(f.film_id) DESC;
92
93 -- Question 2
94 SELECT c.name, count(fc.film_id)
95 FROM category c JOIN film_category fc
96 ON c.category_id = fc.category_id
97 GROUP BY fc.category_id
98 ORDER BY count(fc.film_id) DESC;
99
100 -- Question 3
101 SELECT c.first_name, c.last_name, count(r.rental_id) AS pg_films_at_299
102 FROM customer c JOIN rental r ON c.customer_id = r.customer_id
103 JOIN inventory i ON r.inventory_id = i.inventory_id
104 JOIN payment p ON p.rental_id = r.rental_id
105 JOIN film f ON i.film_id = f.film_id
106 WHERE f.rating = "PG" AND p.amount = 2.99
107 GROUP BY c.customer_id
108 HAVING count(r.rental_id) >= 4
109 ORDER BY count(r.rental_id) DESC;
110
111 -- Question 4
112 SELECT f.title, max(p.amount) AS max_rental_payment_for_movie
113 FROM rental r JOIN inventory i ON r.inventory_id = i.inventory_id
114 JOIN film f ON f.film_id = i.film_id
115 JOIN payment p ON p.rental_id = r.rental_id
116 WHERE f.rating = "G" AND p.amount = (
117     SELECT max(p.amount)
118     FROM rental r JOIN inventory i ON
119     r.inventory_id = i.inventory_id
120     JOIN film f ON f.film_id = i.film_id
121     JOIN payment p ON p.rental_id = r.rental_id
122 )
123 GROUP BY f.film_id;
124
125 -- Question 5
126 SELECT c.name, count(f.film_id)
127 FROM category c JOIN film_category fc ON c.category_id = fc.category_id
128 JOIN film f ON fc.film_id = f.film_id
129 WHERE f.rating = "PG"
130 GROUP BY c.category_id
131 HAVING count(f.film_id) = (SELECT max(val) AS max_films FROM
132     (SELECT count(f.film_id) AS val
133     FROM category c JOIN film_category fc ON c.category_id
134     = fc.category_id
135     JOIN film f ON fc.film_id = f.film_id
136     WHERE f.rating = "PG"
137     GROUP BY c.category_id)
138     maximum)
139 ORDER BY count(f.film_id) DESC;

```

```

137
138 -- Question 6
139 SELECT f.title, count(r.rental_id)
140 FROM film f JOIN inventory i ON i.film_id = f.film_id
141 JOIN rental r ON r.inventory_id = i.inventory_id
142 WHERE f.rating = "G"
143 GROUP BY f.film_id
144 HAVING count(r.rental_id) > (
145     SELECT avg(val) AS max_times_rented
146     FROM (
147         SELECT count(r.rental_id) AS val
148         FROM film f JOIN inventory i ON i.film_id =
149             f.film_id
150         JOIN rental r ON r.inventory_id = i.inventory_id
151         WHERE f.rating = "G"
152         GROUP BY f.film_id)
153     t)
154 ORDER BY count(r.rental_id) DESC;
155
156 -- Question 7
157 SELECT DISTINCT a1.first_name, a1.last_name
158 FROM film f JOIN film_actor fa ON f.film_id = fa.film_id
159 JOIN actor a1 ON a1.actor_id = fa.actor_id
160 WHERE NOT EXISTS (
161     SELECT a2.actor_id
162     FROM film f JOIN film_actor fa ON f.film_id = fa.film_id
163     JOIN actor a2 ON fa.actor_id = a2.actor_id
164     WHERE f.rating = "G" AND a1.actor_id = a2.actor_id);
165
166 -- Question 8
167 -- was unsure how to do this one
168
169 -- Question 9
170 SELECT DISTINCT a.actor_id, a.first_name, a.last_name, (SELECT count(g_films)
171     FROM (SELECT count(fa1.film_id)
172         AS g_films
173     FROM film f1 JOIN film_actor fa1 ON f1.film_id = fa1.film_id
174     JOIN actor a1 ON a1.actor_id = fa1.actor_id
175     WHERE f1.rating = "G"
176     GROUP BY a1.actor_id) g_films_count)/(SELECT count(all_films)
177     FROM (SELECT count(fa2.film_id)
178         AS all_films
179     FROM film f2 JOIN film_actor fa2 ON f2.film_id = fa2.film_id
180     JOIN actor a2 ON a2.actor_id = fa2.actor_id
181     GROUP BY a2.actor_id) all_films_count) AS percentage_g_movies
182 FROM film f JOIN film_actor fa ON f.film_id = fa.film_id
183 JOIN actor a ON a.actor_id = fa.actor_id
184 GROUP BY a.actor_id;
185
186 -- Question 10
187 SELECT f.title
188 FROM film f LEFT JOIN film_actor fa ON f.film_id = fa.film_id
189 WHERE fa.actor_id IS NULL;
190
191 -- Question 11
192 SELECT f.title
193 FROM film f JOIN inventory i ON f.film_id = i.film_id
194 LEFT JOIN rental r ON i.inventory_id = r.inventory_id
195 WHERE r.rental_id IS NULL;
196
197 -- Question 12
198 SELECT film_id, count(*)
199 FROM (
200     SELECT DISTINCT f.film_id, a.actor_id
201     FROM film f JOIN film_actor fa ON f.film_id = fa.film_id
202     LEFT JOIN actor a ON a.actor_id = fa.actor_id) faq
203 GROUP BY faq.film_id
204 ORDER BY count(*);

```