UDACITY

‹ Return to "Programming for Data Science with Python" in the classroom

Investigate a Relational Database

REVIEW

CODE REVIEW

HISTORY

▼ SQL QUERY - DVD RENTAL PROJECT - ONYEKABA NZUBECHUKWU JUDE.txt

```
1
2  /* Query 1 -  How many films rented was returned on time, early, or late and t
3
4  WITH t1 AS (
5  SELECT r.inventory_id, p.amount payment,
6  DATE_PART('day',
7  r.return_date - r.rental_date) date_dif
8  FROM rental r
9  JOIN payment p
10 ON r.rental_id = p.rental_id),
11
12 t2 AS (
13 SELECT payment, t1.date_dif, f.rental_duration,
14 CASE WHEN date_dif < rental_duration THEN 'Early Return'
15 WHEN date_dif > rental_duration THEN 'Late Return'
16 ELSE 'On-Time Return'
17 END AS status
18 FROM film f
19 JOIN inventory i
20 ON f.film_id = i.film_id
21 JOIN t1
22 ON t1.inventory_id = i.inventory_id)
23
24 SELECT Status, COUNT(*) Number, SUM(payment) Total
25 FROM t2
26 GROUP BY 1
27 ORDER BY 2 DESC
28
```

```sql
/* Query 2 - Top and Bottom 10 customers and their details*/

     WITH t1 AS (
     SELECT *
     FROM payment p
     JOIN customer c
     ON p.customer_id = c.customer_id
     JOIN address a
     ON c.address_id = a.address_id
     ),

   t2 AS (
   SELECT t1.first_name || ' ' || t1.last_name  Customer, SUM(t1.amount) Total
   FROM t1
   GROUP BY 1,3,4,5),

    t3 AS (
    SELECT *, ROW_NUMBER() OVER (ORDER BY t2.Total_Payment DESC)  TopFive, ROW
   FROM t2)

SELECT customer, total_payment, address, password
FROM t3
WHERE TopFive <= 10 or BottomFive <= 10


/* Query 3 - Movie Distribution per Category by Quartile*/
WITH t1 AS(
           SELECT f.title, c.name , f.rental_duration, NTILE(4) OVER (ORDER BY
           FROM film_category fc
           JOIN category c
           ON c.category_id = fc.category_id
           JOIN film f
           ON f.film_id = fc.film_id
           GROUP BY 1, 2, 3
           ORDER BY 1, 2)
SELECT t1.name, t1.standard_quartile, COUNT(t1.standard_quartile)
FROM t1
GROUP BY 1, 2
ORDER BY 1, 2


/ * Query 4 - Actors name in top 10 rented films*/
WITH t1 AS (
         SELECT *
         FROM inventory i
         JOIN rental r
         ON i.inventory_id = r.inventory_id
         JOIN payment p
         ON r.rental_id = p.rental_id
         JOIN film f
         ON i.film_id = f.film_id
         JOIN film_actor fa
         ON f.film_id = fa.film_id
         JOIN actor a
         ON fa.actor_id = a.actor_id
         ),

    t2 AS(
        SELECT t1.first_name || '' ||t1.last_name Actor_Name, SUM(t1.amount) T
       FROM t1
       GROUP BY 1
```

```
            ),
    t3 AS (
       SELECT *, ROW_NUMBER() OVER (ORDER BY t2.Total_Amount DESC) TopFive, RO
       FROM t2)
SELECT Actor_Name, Total_Amount, Number_of_Rents
FROM t3
WHERE TopFive <= 10 OR BottomFive <= 10
```

RETURN TO PATH