Homework 2

New Attempt

Due Oct 23 by 11:59pm **Points** 60 **Submitting** a file upload

1. Write SQL Queries corresponding to the following questions (10 points each)

(a) Using the ag_class dataset: For employees who have dependents - list the employee name and country (-ies) of their departments

(b) Using the airbnb_listings dataset: Find the frequency distribution of (property-type, room-type) pairs for listings that have a gym and free parking

(c) Using the airbnb_reviews dataset: For comments that are "automated posting"s about cancellations, find the number of days before arrival that the reservation was cancelled.

2. Let's assume FUNDS is a single-column table of mutual fund symbols and INVESTS is the table (InvestorName, Fund, InvestmentDate, Amount). The intention of the following query is to find funds that no one has invested in yet. (10 points)

```
SELECT Funds.symbol

FROM Funds F

WHERE NOT EXISTS(SELECT *
          FROM Invests I
          WHERE F.Symbol = I.fund);
```

However the query returns a NULL although there are several unutilized funds in the Funds table. When can this happen? How can you fix the problem?

3. Consider the following tables

Subscribers

SubscriberNumber	Name
1	Mary Golden
2	Fred Steward
3	Jeff Rose
4	Anne Frederick

Purchases

OrderNumber	Date	Subscriber	Amount
1	2021-03-21	1	150.00
2	2021-04-15	2	250.00
3	2020-10-03	1	180.00
4	2020-09-18	3	110:00
5	2021--11-22	1	300.00

Intention of the query: retrieve a list of all subscribers, and the total amount they have spent since the beginning of year 2021.

Actual query:

```
SELECT S.name, SUM(amount) as Total
FROM Subscribers S
LEFT OUTER JOIN Purchases P
  ON S.subscribernumber = P.subscriber
WHERE P.date >= '2021-01-01'::date
GROUP BY S.name;
```

Is this query correct? If not, state why it is not and write the correct query. (15 points)

4. What is a correlated subquery? (5 points)

ASSIGNMENT-02

1. (a)

```
select concat(emp.first_name,' ',emp.last_name) as employee_name, c.country_name from
ag_class.ag_class_employees emp join ag_class.ag_class_dependents d on emp.employee_id
= d.employee_id join ag_class.ag_class_departments d2 on emp.department_id =
d2.department_id join ag_class.ag_class_locations l on d2.location_id = l.location_id join
ag_class.ag_class_countries c on c.country_id = l.country_id where d.employee_id is not
null;
```

(b)

```
SELECT v.property_type, v.room_type,
COUNT(*) AS AbsFreq,
CAST(ROUND(100. * (COUNT(*)) /
  (SELECT COUNT(*) FROM "Airbnb"."Listings"), 0) AS INT) AS AbsPerc
FROM "Airbnb"."Listings" AS v
where v.amenities ilike '%Free%Parking%' and v.amenities ilike '%gym%'
GROUP BY v.property_type, v.room_type;
```

Assumption : The amenities column will provide us with an exhaustive list of the spaces containing both Free Parking and Gym

(c)

```
SELECT NULLIF(regexp_replace(comments, '\D','','g'), '')::numeric AS
no_of_days FROM "Airbnb"."Reviews" where comments ilike '%automated
posting%' ;
```

Assumption : The comments containing the expression ‘automated posting’ will also contain the number of days before which the cancellation was done.

2. This can happen when there are 0s in Amount . So to fix the problem , we can do zero check for Amount column in the subquery.

   SELECT Funds.symbol
   FROM Funds F
   WHERE NOT EXISTS(SELECT * FROM Invests WHERE F.Symbol = I.fund AND I.Amount != 0);

3.
This query is incorrect because it will not show the list of all the subscribers. The query would give the following result:

Name	Total
Mary Golden	450.00
Fred Steward	250.00

But we need the following result:

Name	Total

Mary Golden	450.00
Fred Steward	250.00
Jeff Rose	0.00
Anne Frederick	0.00

So, the correct query is –

```
SELECT S.name, SUM(amount) as Total
FROM Subscribers S
         LEFT OUTER JOIN (select * from Purchases P where P.date >= '2021-01-01'::date) P2
                         ON S.subscribernumber = P2.subscriber
GROUP BY S.name;
```

4.

A correlated subquery is a subquery that contains a reference to a table (in the parent query) that also appears in the outer query. PostgreSQL evaluates from inside to outside.

Correlated subquery syntax:

SELECT column1, column2, ….

FROM table1 outerr

WHERE column1 operator (SELECT column1, column2 FROM table2 WHERE expr1 = outer.expr2)

Example - 1: PostgreSQL Correlated Subqueries

Following query finds all employees who earn more than the average salary in their department.

Code:

```
SELECT last_name, salary, department_id

FROM employees outerr

WHERE salary>

(SELECT AVG(salary)

FROM employees

WHERE department_id = outerr.department_id);
```