

MYSQL Assignment

1. Write a query to display the policytypeid, policytypename, description of all the car's policy details.

```
SELECT pst.policy_type_id, rpt.policy_type_name, pst.description  
FROM policy_sub_types pst  
JOIN ref_policy_types rpt ON pst.policy_type_code = rpt.policy_type_code  
WHERE rpt.policy_type_name = 'car';
```

2. Write a query to display the policytypecode, no of policies in each code with alias name NO_OF_POLICIES.

```
SELECT rpt.policy_type_code, COUNT(up.policy_no) AS NO_OF_POLICIES  
FROM user_policies up  
JOIN policy_sub_types pst ON up.policy_type_id = pst.policy_type_id  
JOIN ref_policy_types rpt ON pst.policy_type_code = rpt.policy_type_code  
GROUP BY rpt.policy_type_code;
```

3. Write a query to display the userid, firstname, lastname, email, mobileno who are residing in Chennai.

```
SELECT u.user_id, u.firstname, u.lastname, u.email, u.mobileno  
FROM user_details u  
JOIN address_details a ON u.address_id = a.address_id  
WHERE a.city = 'Chennai';
```

4. Write a query to display the userid, firstname lastname with alias name USER_NAME, email, mobileno who has taken the car policies.

```
SELECT DISTINCT u.user_id, CONCAT(u.firstname, ' ', u.lastname) AS  
USER_NAME, u.email, u.mobileno
```

```
FROM user_details u
JOIN user_policies up ON u.user_id = up.user_id
JOIN policy_sub_types pst ON up.policy_type_id = pst.policy_type_id
JOIN ref_policy_types rpt ON pst.policy_type_code = rpt.policy_type_code
WHERE rpt.policy_type_name = 'car';
```

5. Write a query to display the userid, firstname, lastname who has taken the car policies but not home policies.

```
SELECT u.user_id, u.firstname, u.lastname
FROM user_details u
WHERE u.user_id IN (
    SELECT up.user_id
    FROM user_policies up
    JOIN policy_sub_types pst ON up.policy_type_id = pst.policy_type_id
    JOIN ref_policy_types rpt ON pst.policy_type_code = rpt.policy_type_code
    WHERE rpt.policy_type_name = 'car'
) AND u.user_id NOT IN (
    SELECT up.user_id
    FROM user_policies up
    JOIN policy_sub_types pst ON up.policy_type_id = pst.policy_type_id
    JOIN ref_policy_types rpt ON pst.policy_type_code = rpt.policy_type_code
    WHERE rpt.policy_type_name = 'home'
);
```

6. Write a query to display the policytypecode, policytype name which policytype has maximum no of policies.

```
SELECT rpt.policy_type_code, rpt.policy_type_name
```

```
FROM ref_policy_types rpt
JOIN policy_sub_types pst ON rpt.policy_type_code = pst.policy_type_code
JOIN user_policies up ON pst.policy_type_id = up.policy_type_id
GROUP BY rpt.policy_type_code, rpt.policy_type_name
ORDER BY COUNT(up.policy_no) DESC
LIMIT 1;
```

7. Write a query to display the userid, firstname, lastname, city state whose city is ending with 'bad'.

```
SELECT u.user_id, u.firstname, u.lastname, a.city, a.state
FROM user_details u
JOIN address_details a ON u.address_id = a.address_id
WHERE a.city LIKE '%bad';
```

8. Write a query to display the userid, firstname, lastname, policymno, dateregistered who has registered before may 2012.

```
SELECT u.user_id, u.firstname, u.lastname, up.policy_no, up.date_registered
FROM user_details u
JOIN user_policies up ON u.user_id = up.user_id
WHERE up.date_registered < '2012-05-01';
```

9. Write a query to display the userid, firstname, lastname who has taken more than one policies.

```
SELECT u.user_id, u.firstname, u.lastname
FROM user_details u
JOIN user_policies up ON u.user_id = up.user_id
GROUP BY u.user_id, u.firstname, u.lastname
```

HAVING COUNT(up.policy_no) > 1;

10. Write a query to display the policytypecode, policytypename, policytypeid, userid, policyno whose maturity will fall in the month of august 2013.

```
/* Assuming maturity is date_registered + maturityperiod (in years) */

SELECT rpt.policy_type_code, rpt.policy_type_name, pst.policy_type_id,
       up.user_id, up.policy_no
  FROM user_policies up
  JOIN policy_sub_types pst ON up.policy_type_id = pst.policy_type_id
  JOIN ref_policy_types rpt ON pst.policy_type_code = rpt.policy_type_code
 WHERE pst.maturityperiod IS NOT NULL
   AND DATE_ADD(up.date_registered, INTERVAL pst.maturityperiod YEAR)
      BETWEEN '2013-08-01' AND '2013-08-31';
```

11. Write a query to display the policytypecode, policytypename, policytypeid whose maturity amount is the double than the total paid amount.

```
SELECT rpt.policy_type_code, rpt.policy_type_name, pst.policy_type_id
  FROM policy_sub_types pst
  JOIN ref_policy_types rpt ON pst.policy_type_code = rpt.policy_type_code
 WHERE pst.maturityamount > 2 * (CASE WHEN pst.yearsofpayments IS NOT
        NULL THEN (pst.amount * pst.yearsofpayments) ELSE pst.amount END);
```

12. Write a query to display the userid, total amount paid by the customer with alias name total_amount.

```
SELECT user_id, SUM(amount) AS total_amount
  FROM policy_payments
 GROUP BY user_id;
```

13. Write a query to display the user_id, policy_no, total amount paid by the customer for each policies.

```
SELECT user_id, policy_no, SUM(amount) AS total_amount  
FROM policy_payments  
GROUP BY user_id, policy_no;
```

14. Write a query to display the user_id, policy_no, balance_amount for each policies.

```
SELECT up.user_id, up.policy_no,  
       (CASE WHEN pst.yearsofpayments IS NOT NULL THEN (pst.amount *  
               pst.yearsofpayments) ELSE pst.amount END) - IFNULL(SUM(pp.amount), 0)  
       AS balance_amount  
FROM user_policies up  
JOIN policy_sub_types pst ON up.policy_type_id = pst.policy_type_id  
LEFT JOIN policy_payments pp ON up.policy_no = pp.policy_no  
GROUP BY up.user_id, up.policy_no, pst.amount, pst.yearsofpayments;
```

15. Write a query to display the user_id, policy_no, balancepayment years with alias name BALANCE_YEARS for all the customer for each policies.

```
SELECT up.user_id, up.policy_no, (pst.yearsofpayments -  
        COUNT(pp.receipno)) AS BALANCE_YEARS  
FROM user_policies up  
JOIN policy_sub_types pst ON up.policy_type_id = pst.policy_type_id  
LEFT JOIN policy_payments pp ON up.policy_no = pp.policy_no  
WHERE pst.yearsofpayments IS NOT NULL  
GROUP BY up.user_id, up.policy_no, pst.yearsofpayments;
```

16. Write a query to display the user details userid, firstname, last who has taken car, home and life policies.

```
SELECT u.user_id, u.firstname, u.lastname  
FROM user_details u  
JOIN user_policies up ON u.user_id = up.user_id  
JOIN policy_sub_types pst ON up.policy_type_id = pst.policy_type_id  
JOIN ref_policy_types rpt ON pst.policy_type_code = rpt.policy_type_code  
WHERE rpt.policy_type_name IN ('car', 'home', 'life')  
GROUP BY u.user_id, u.firstname, u.lastname  
HAVING COUNT(DISTINCT rpt.policy_type_name) = 3;
```

17. Write a query to select policy_type_code, total amount paid by all the customers with alias name total_amount for each policy department.

```
SELECT rpt.policy_type_code, SUM(pp.amount) AS total_amount  
FROM policy_payments pp  
JOIN user_policies up ON pp.policy_no = up.policy_no  
JOIN policy_sub_types pst ON up.policy_type_id = pst.policy_type_id  
JOIN ref_policy_types rpt ON pst.policy_type_code = rpt.policy_type_code  
GROUP BY rpt.policy_type_code;
```

18. Write a query to select user_id, user_name, policy_type_code, policy_type_id of users who has registered more than one policy type under same policy code.

```
SELECT u.user_id, u.firstname AS user_name, pst.policy_type_code,  
pst.policy_type_id  
FROM user_policies up  
JOIN policy_sub_types pst ON up.policy_type_id = pst.policy_type_id  
JOIN user_details u ON up.user_id = u.user_id
```

```

WHERE (up.user_id, pst.policy_type_code) IN (
    SELECT up2.user_id, pst2.policy_type_code
    FROM user_policies up2
    JOIN policy_sub_types pst2 ON up2.policy_type_id = pst2.policy_type_id
    GROUP BY up2.user_id, pst2.policy_type_code
    HAVING COUNT(DISTINCT pst2.policy_type_id) > 1
);

```

19. Write a query to display the policy_type_code, policytype name in which policy department has min number of policies registered.

```

SELECT rpt.policy_type_code, rpt.policy_type_name
FROM ref_policy_types rpt
JOIN policy_sub_types pst ON rpt.policy_type_code = pst.policy_type_code
JOIN user_policies up ON pst.policy_type_id = up.policy_type_id
GROUP BY rpt.policy_type_code, rpt.policy_type_name
ORDER BY COUNT(up.policy_no) ASC
LIMIT 1;

```

20. Write a query to display the user_id, user_name, address, phoneno, policytypecode, policytypeid, policytypename, who has completed all payments for the policies.

```

SELECT u.user_id, u.firstname AS user_name, CONCAT(a.h_no, ', 'a.city, ', 'a.state) AS address, u.mobileno, pst.policy_type_code, pst.policy_type_id, rpt.policy_type_name
FROM user_details u
JOIN address_details a ON u.address_id = a.address_id
JOIN user_policies up ON u.user_id = up.user_id
JOIN policy_sub_types pst ON up.policy_type_id = pst.policy_type_id

```

```
JOIN ref_policy_types rpt ON pst.policy_type_code = rpt.policy_type_code
LEFT JOIN policy_payments pp ON up.policy_no = pp.policy_no
GROUP BY u.user_id, up.policy_no, pst.amount, pst.yearsofpayments
HAVING (CASE WHEN pst.yearsofpayments IS NOT NULL THEN
(pst.amount * pst.yearsofpayments) ELSE pst.amount END) <=
SUM(IFNULL(pp.amount, 0));
```

21. Write a query to display the user_id, user_name, address, phoneno, policytypecode, policytypeid, policytypename, date of register who has registered latest 2.

```
SELECT u.user_id, u.firstname AS user_name, CONCAT(a.h_no, ', 'a.city, ', '
,a.state) AS address, u.mobileno, pst.policy_type_code, pst.policy_type_id,
rpt.policy_type_name, up.date_registered
FROM user_policies up
JOIN user_details u ON up.user_id = u.user_id
JOIN address_details a ON u.address_id = a.address_id
JOIN policy_sub_types pst ON up.policy_type_id = pst.policy_type_id
JOIN ref_policy_types rpt ON pst.policy_type_code = rpt.policy_type_code
ORDER BY up.date_registered DESC
LIMIT 2;
```