DANG NHAT VIET
MySQL

**QUESTION 1**

**a) Show the number of products available for each accepted risk level.**

```
SELECT
        accepted_risk_level,
        COUNT(DISTINCT product_id) AS products

FROM Products

GROUP BY accepted_risk_level;
```

**b) Show the average interest rate of products provided by HSBC bank.**

```
SELECT
        AVG(interest_rate) AS HSBC_avg_interest_rate

FROM Products

WHERE bank_id IN ( SELECT bank_id
                   FROM Banks
                   WHERE bank_name = "HSBC"
                   );
```

**c) Show 2 banks that have most high risk products.**

```
SELECT

        b.bank_name,

        COUNT(DISTINCT p.product_id) AS products

FROM Products AS p
        LEFT JOIN Banks AS b
                ON p.bank_id = b.bank_id

WHERE p.accepted_risk_level = "high"

GROUP BY b.bank_name

ORDER BY products DESC
LIMIT 2;
```

**d) Show which source brings to the marketplace more low risk customers.**

```
SELECT
        source,
        COUNT(DISNTINCT customer_id) AS low_risk_customers


FROM Customers


WHERE estimated_risk_level = “low”


GROUP BY source


ORDER BY low_risk_customers DESC;
```

**e) Show all months of the year 2017 that the number of customers applying for loans are 20% higher than the monthly average number of customers of the year.**

```
SELECT
        MONTH(apply_date) AS month,
        COUNT(DISTINCT customer_id) AS customers


FROM Leads


WHERE YEAR(apply_date) = 2017


GROUP BY month


HAVING customers / (SELECT
                        COUNT(DISTINCT customer_id)/12
                    FROM Leads
                    WHERE YEAR(apply_date) = 2017
                    ) > 1.2;
```

**f) Show the names of all leads who applied in 2017 and are older than 90% of all leads who applied in 2016**

```
-- Create a temporary table to rank customer age in 2016

CREATE TEMPORARY TABLE age_percent_rank_2016
SELECT
        customer_age,
        ROUND(
                PERCENT_RANK() OVER (
                        ORDER BY age
                        )
                ,2) AS age_percentile_rank
FROM Customers
WHERE customer_id IN (  SELECT customer_id
                        FROM Leads
                        WHERE YEAR(apply_date) = 2016
                        );


-- Filter all customers who meet all conditions
SELECT
        customer_name,
        customer_age
FROM Customers
WHERE customer_id IN (  SELECT customer_id
                        FROM Leads
                        WHERE YEAR(apply_date) = 2017
                        )
        AND customer_age > (    SELECT
                                        MIN (customer_age)
                                FROM age_percent_rank_2016
                                WHERE age_percentile_rank >= 0.9
```

```
GROUP BY customer_age
);
```