// To find level 1 student count

const countsOflevel1 = async (req, res) => {

  try {

    const { tableName } = req.params;

    const resultQuery = `SELECT \* FROM max\_marks\_for\_each\_co WHERE Main\_Table\_Name = '${tableName}'`;

    const result = await pool.query(resultQuery);

    const firstResult=result[0][0];

    let sql = `SELECT

      SUM(CASE WHEN (\`UT1-Q1\` IS NOT NULL AND \`UT1-Q1\` >=(${firstResult['CO-1']}\*40/100) ) THEN 1 ELSE 0 END) AS sum\_q11,

      SUM(CASE WHEN (\`UT1-Q2\` IS NOT NULL AND \`UT1-Q2\` >=(${firstResult['CO-2']}\*40/100) ) THEN 1 ELSE 0 END) AS sum\_q12,

      SUM(CASE WHEN (\`UT2-Q1\` IS NOT NULL AND \`UT2-Q1\` >=(${firstResult['CO-3']}\*40/100) ) THEN 1 ELSE 0 END) AS sum\_q21,

      SUM(CASE WHEN (\`UT2-Q2\` IS NOT NULL AND \`UT2-Q2\` >=(${firstResult['CO-4']}\*40/100) ) THEN 1 ELSE 0 END) AS sum\_q22,

      SUM(CASE WHEN (\`UT3-Q1\` IS NOT NULL AND \`UT3-Q1\` >=(${firstResult['CO-5']}\*40/100) ) THEN 1 ELSE 0 END) AS sum\_q31,

      SUM(CASE WHEN (\`UT3-Q2\` IS NOT NULL AND \`UT3-Q2\` >=(${firstResult['CO-6']}\*40/100) ) THEN 1 ELSE 0 END) AS sum\_q32,

      SUM(CASE WHEN (UA IS NOT NULL AND UA >=(100\*40/100) ) THEN 1 ELSE 0 END) AS sum\_UA

      FROM ${tableName} AS new\_table;`;

    const queryResult = await pool.query(sql);

    if (queryResult.length === 0) {

      console.log("No data found");

      res.status(404).send("No data found");

      return;

    }

    // Log the entire first row of the result

    console.log("First row of the result:", queryResult[0]);

    // Attempt to fetch the value of sum\_q11

    const present = queryResult[0];

    // console.log("Fetched sum\_q11:", present);

    res.send(present); // Sending a JSON response with the value of sum\_q11

  } catch (error) {

    console.error("Error fetching data:", error);

    res.status(500).send("Internal Server Error");

  }

};

// To find level 2 student count

const countsOflevel2 = async (req, res) => {

  try {

    const { tableName } = req.params;

    const resultQuery = `SELECT \* FROM max\_marks\_for\_each\_co WHERE Main\_Table\_Name = '${tableName}'`;

    const result = await pool.query(resultQuery);

    const firstResult=result[0][0];

    let sql = `SELECT

      SUM(CASE WHEN (\`UT1-Q1\` IS NOT NULL AND \`UT1-Q1\` >=(${firstResult['CO-1']}\*60/100) ) THEN 1 ELSE 0 END) AS sum\_q11,

      SUM(CASE WHEN (\`UT1-Q2\` IS NOT NULL AND \`UT1-Q2\` >=(${firstResult['CO-2']}\*60/100) ) THEN 1 ELSE 0 END) AS sum\_q12,

      SUM(CASE WHEN (\`UT2-Q1\` IS NOT NULL AND \`UT2-Q1\` >=(${firstResult['CO-3']}\*60/100) ) THEN 1 ELSE 0 END) AS sum\_q21,

      SUM(CASE WHEN (\`UT2-Q2\` IS NOT NULL AND \`UT2-Q2\` >=(${firstResult['CO-4']}\*60/100) ) THEN 1 ELSE 0 END) AS sum\_q22,

      SUM(CASE WHEN (\`UT3-Q1\` IS NOT NULL AND \`UT3-Q1\` >=(${firstResult['CO-5']}\*60/100) ) THEN 1 ELSE 0 END) AS sum\_q31,

      SUM(CASE WHEN (\`UT3-Q2\` IS NOT NULL AND \`UT3-Q2\` >=(${firstResult['CO-6']}\*60/100) ) THEN 1 ELSE 0 END) AS sum\_q32,

      SUM(CASE WHEN (UA IS NOT NULL AND UA >=(100\*60/100) ) THEN 1 ELSE 0 END) AS sum\_UA

      FROM ${tableName} AS new\_table;`;

    const queryResult = await pool.query(sql);

    if (queryResult.length === 0) {

      console.log("No data found");

      res.status(404).send("No data found");

      return;

    }

    // Log the entire first row of the result

    console.log("First row of the result:", queryResult[0]);

    // Attempt to fetch the value of sum\_q11

    const present = queryResult[0];

    // console.log("Fetched sum\_q11:", present);

    res.send(present); // Sending a JSON response with the value of sum\_q11

  } catch (error) {

    console.error("Error fetching data:", error);

    res.status(500).send("Internal Server Error");

  }

};

// To find level 3 student count

const countsOflevel3 = async (req, res) => {

  try {

    const { tableName } = req.params;

    const resultQuery = `SELECT \* FROM max\_marks\_for\_each\_co WHERE Main\_Table\_Name = '${tableName}'`;

    const result = await pool.query(resultQuery);

    const firstResult=result[0][0];

    let sql = `SELECT

      SUM(CASE WHEN (\`UT1-Q1\` IS NOT NULL AND \`UT1-Q1\` >=(${firstResult['CO-1']}\*66/100) ) THEN 1 ELSE 0 END) AS sum\_q11,

      SUM(CASE WHEN (\`UT1-Q2\` IS NOT NULL AND \`UT1-Q2\` >=(${firstResult['CO-2']}\*66/100) ) THEN 1 ELSE 0 END) AS sum\_q12,

      SUM(CASE WHEN (\`UT2-Q1\` IS NOT NULL AND \`UT2-Q1\` >=(${firstResult['CO-3']}\*66/100) ) THEN 1 ELSE 0 END) AS sum\_q21,

      SUM(CASE WHEN (\`UT2-Q2\` IS NOT NULL AND \`UT2-Q2\` >=(${firstResult['CO-4']}\*66/100) ) THEN 1 ELSE 0 END) AS sum\_q22,

      SUM(CASE WHEN (\`UT3-Q1\` IS NOT NULL AND \`UT3-Q1\` >=(${firstResult['CO-5']}\*66/100) ) THEN 1 ELSE 0 END) AS sum\_q31,

      SUM(CASE WHEN (\`UT3-Q2\` IS NOT NULL AND \`UT3-Q2\` >=(${firstResult['CO-6']}\*66/100) ) THEN 1 ELSE 0 END) AS sum\_q32,

      SUM(CASE WHEN (UA IS NOT NULL AND UA >=(100\*66/100) ) THEN 1 ELSE 0 END) AS sum\_UA

      FROM ${tableName} AS new\_table;`;

    const queryResult = await pool.query(sql);

    if (queryResult.length === 0) {

      console.log("No data found");

      res.status(404).send("No data found");

      return;

    }

    // Log the entire first row of the result

    console.log("First row of the result:", queryResult[0]);

    // Attempt to fetch the value of sum\_q11

    const present = queryResult[0];

    // console.log("Fetched sum\_q11:", present);

    res.send(present); // Sending a JSON response with the value of sum\_q11

  } catch (error) {

    console.error("Error fetching data:", error);

    res.status(500).send("Internal Server Error");

  }

};