```
-- Assuming your database is named 'soulvibe' and the table is 'CollegeCourses'
USE soulvibe;

-- Q1. Find the top 5 districts with the highest number of colleges offering professional courses.
SELECT District,
    COUNT(DISTINCT `College Name`) AS NumberOfColleges
FROM CollegeCourses
WHERE `Is Professional` = 'Professional Course'
GROUP BY District
ORDER BY NumberOfColleges DESC
LIMIT 5;

-- Q2. Calculate the average course duration (in months) for each Course Type and sort them in descending order.
SELECT `Course Type`,
    AVG(`Course Duration (In months)`) AS AverageCourseDurationMonths
FROM CollegeCourses
GROUP BY `Course Type`
ORDER BY AverageCourseDurationMonths DESC;

-- Q3. Count how many unique College Names offer each Course Category.
SELECT `Course Category`,
    COUNT(DISTINCT `College Name`) AS NumberOfUniqueColleges
FROM CollegeCourses
GROUP BY `Course Category`;

-- Q4. Find the names of colleges offering both Post Graduate and Under Graduate courses.
SELECT `College Name`
```

```
FROM CollegeCourses
WHERE `Course Type` IN ('Post Graduate Course', 'Under Graduate Course')
GROUP BY `College Name`
HAVING COUNT(DISTINCT `Course Type`) = 2;

-- Q5. List all universities that have more than 10 unaided courses that are not professional.
SELECT University
FROM CollegeCourses
WHERE `Course (Aided / Unaided)` = 'Unaided'
  AND `Is Professional` = 'Non-Professional Course'
GROUP BY University
HAVING COUNT(*) > 10;

-- Q6. Display colleges from the "Engineering" category that have at least one course with a duration greater than the category’s average.
SELECT DISTINCT `College Name`
FROM CollegeCourses
WHERE `Course Category` = 'Engineering'
  AND `Course Duration (In months)` >
    (SELECT AVG(`Course Duration (In months)`)
     FROM CollegeCourses
     WHERE `Course Category` = 'Engineering');

-- Q7. Assign a rank to each course within a College Name based on course duration, longest first.
SELECT `College Name`,
       `Course Name`,
       `Course Duration (In months)`,
```

```
    RANK() OVER (PARTITION BY `College Name` ORDER BY `Course Duration (In months)` DESC) AS CourseDurationRank
FROM CollegeCourses;

-- Q8. Find colleges where the longest and shortest course durations are more than 24 months apart.
SELECT `College Name`
FROM CollegeCourses
GROUP BY `College Name`
HAVING (MAX(`Course Duration (In months)`) - MIN(`Course Duration (In months)`)) > 24;

-- Q9. Show the cumulative number of professional courses offered by each university sorted alphabetically.
-- MySQL 8.0+ supports window functions like SUM() OVER().
SELECT University,
    `Course Name`,
    `Is Professional`,
    SUM(CASE WHEN `Is Professional` = 'Professional Course' THEN 1 ELSE 0 END) OVER (PARTITION BY University ORDER BY `College Name`, `Course Name`) AS CumulativeProfessionalCoursesPerUniversity
FROM CollegeCourses
WHERE `Is Professional` = 'Professional Course'
ORDER BY University ASC, `College Name`, `Course Name`;

-- Q10. Using a self-join or CTE, find colleges offering more than one course category.
WITH CollegeCategoryCounts AS (
  SELECT `College Name`,
      COUNT(DISTINCT `Course Category`) AS NumCourseCategories
```

```
    FROM CollegeCourses
    GROUP BY `College Name`
)
SELECT `College Name`
FROM CollegeCategoryCounts
WHERE NumCourseCategories > 1;


-- Q11. Create a temporary table (CTE) that includes average duration of courses by district and use it to list talukas where the average course duration is above the district average.
WITH DistrictAverage AS (
    SELECT District,
        AVG(`Course Duration (In months)`) AS AvgDurationDistrict
    FROM CollegeCourses
    GROUP BY District
),
TalukaAverage AS (
    SELECT District,
        Taluka,
        AVG(`Course Duration (In months)`) AS AvgDurationTaluka
    FROM CollegeCourses
    GROUP BY District, Taluka
)
SELECT TA.District,
    TA.Taluka,
    TA.AvgDurationTaluka,
    DA.AvgDurationDistrict
FROM TalukaAverage AS TA
JOIN DistrictAverage AS DA
```

```
  ON TA.District = DA.District
WHERE TA.AvgDurationTaluka > DA.AvgDurationDistrict;


-- Q12. Create a new column classifying course duration as: Short (< 12 months), Medium (12-36 months), Long (> 36 months). Then count the number of each duration type per course category.
SELECT `Course Category`,
    CASE
        WHEN `Course Duration (In months)` < 12 THEN 'Short'
        WHEN `Course Duration (In months)` >= 12 AND `Course Duration (In months)` <= 36 THEN 'Medium'
        WHEN `Course Duration (In months)` > 36 THEN 'Long'
        ELSE 'Unknown'
    END AS CourseDurationClassification,
    COUNT(*) AS NumberOfCourses
FROM CollegeCourses
GROUP BY `Course Category`,
      CASE
        WHEN `Course Duration (In months)` < 12 THEN 'Short'
        WHEN `Course Duration (In months)` >= 12 AND `Course Duration (In months)` <= 36 THEN 'Medium'
        WHEN `Course Duration (In months)` > 36 THEN 'Long'
        ELSE 'Unknown'
      END
ORDER BY `Course Category`, CourseDurationClassification;


-- Q13. Extract only the course specialization from Course Name. (e.g., from "Bachelor of Engineering (B. E.) - Electrical", extract "Electrical")
-- This query uses SUBSTRING_INDEX for MySQL.
```

```
SELECT `Course Name`,
    TRIM(SUBSTRING_INDEX(`Course Name`, ' - ', -1)) AS CourseSpecialization
FROM CollegeCourses
WHERE `Course Name` LIKE '% - %';

-- Q14. Count how many courses include the word Engineering in the name.
SELECT COUNT(*) AS NumberOfEngineeringCourses
FROM CollegeCourses
WHERE `Course Name` LIKE '%Engineering%';

-- Q15. List all unique combinations of Course Name, Course Type, and Course Category.
SELECT DISTINCT `Course Name`,
        `Course Type`,
        `Course Category`
FROM CollegeCourses;

-- Q16. Write a query to get all courses that are not offered by any Government college.
SELECT DISTINCT `Course Name`, `College Name`
FROM CollegeCourses
WHERE `College Type` != 'Government';

-- Q17. Find the university that has the second-highest number of aided courses.
-- This uses a subquery with LIMIT and OFFSET for MySQL.
SELECT University, COUNT(*) AS NumberOfAidedCourses
FROM CollegeCourses
WHERE `Course (Aided / Unaided)` = 'Aided'
GROUP BY University
ORDER BY NumberOfAidedCourses DESC
```

```
LIMIT 1 OFFSET 1; -- LIMIT 1 gets one row, OFFSET 1 skips the first (highest)


-- Q18. Show courses whose durations are above the median course duration.
-- MySQL doesn't have a direct MEDIAN function like PERCENTILE_CONT.
-- This approach calculates the median using ROW_NUMBER and COUNT.
WITH RankedDurations AS (
   SELECT `Course Duration (In months)`,
        ROW_NUMBER() OVER (ORDER BY `Course Duration (In months)`) AS rn,
        COUNT(*) OVER () AS total_rows
   FROM CollegeCourses
)
SELECT C.`Course Name`,
     C.`Course Duration (In months)`
FROM CollegeCourses C
WHERE C.`Course Duration (In months)` > (
   SELECT AVG(`Course Duration (In months)`)
   FROM RankedDurations
   WHERE rn IN (FLOOR((total_rows + 1) / 2), CEIL((total_rows + 1) / 2))
);


-- Q19. For each University, find the percentage of unaided courses that are
professional.SELECT University,
     (SUM(CASE WHEN `Course (Aided / Unaided)` = 'Unaided' AND `Is Professional` =
'Professional Course' THEN 1 ELSE 0 END) * 100.0) /
     NULLIF(SUM(CASE WHEN `Course (Aided / Unaided)` = 'Unaided' THEN 1 ELSE 0 END), 0)
AS PercentageProfessionalUnaidedCourses
FROM CollegeCourses
GROUP BY University
ORDER BY University;
```

-- Q20. Determine which Course Category has the highest average course duration and display the top 3.

```
SELECT `Course Category`,
    AVG(`Course Duration (In months)`) AS AverageCourseDuration
FROM CollegeCourses
GROUP BY `Course Category`
ORDER BY AverageCourseDuration DESC
LIMIT 3;
```