SQL Assignment: Level Up Your Database Skills!

Let's start by setting up our database, reviewing the data it holds, and then exploring some fundamental SQL concepts! 👇

Database Setup and Data Insertion

```sql
DROP DATABASE IF EXISTS Assignment__Queries;
CREATE DATABASE Assignment__Queries;
USE Assignment__Queries;

-- Create Tables
CREATE TABLE programmer(
    pname VARCHAR(20) PRIMARY KEY,
    dob DATE,
    doj DATE,
    gender VARCHAR(1),
    prof1 VARCHAR(20),
    prof2 VARCHAR(20),
    salary INT
);

CREATE TABLE software(
    pname VARCHAR(20),
    title VARCHAR(20),
    developin VARCHAR(20),
    scost DECIMAL(10, 2), -- Changed to DECIMAL for more accurate cost representation
    dcost DECIMAL(10, 2), -- Changed to DECIMAL for more accurate cost representation
    sold INT,
    FOREIGN KEY(pname) REFERENCES programmer(pname) ON DELETE CASCADE
);

CREATE TABLE studies(
    pname VARCHAR(20) PRIMARY KEY,
    institute VARCHAR(20),
    course VARCHAR(20),
    coursefee INT,
    FOREIGN KEY(pname) REFERENCES programmer(pname) ON DELETE CASCADE
);

-- Insert Data
INSERT INTO programmer VALUES('anand', '1966-04-12','1992-04-21','m','pascal','basic',3200);
INSERT INTO programmer VALUES('altaf', '1964-07-02','1990-11-13','m','clipper','cobol',2800);
INSERT INTO programmer VALUES('juliana','1960-01-31','1990-04-21','f','cobol','dbase',3000);
INSERT INTO programmer VALUES('kamala', '1968-10-30','1992-01-02','f','c','dbase',2900);
INSERT INTO programmer VALUES('mary', '1970-06-24','1991-02-01','f','cpp','oracle', 4500);
INSERT INTO programmer VALUES('nelson', '1985-09-11','1989-10-11','m','cobol','dbase',2500);
INSERT INTO programmer VALUES('pattrick','1965-11-10','1990-04-
21','m','pascal','clipper',2800);
INSERT INTO programmer VALUES('qadir', '1965-08-31','1991-04-21','m','assembly', 'c',3000);
INSERT INTO programmer VALUES('ramesh', '1967-05-03','1991-02-
28','m','pascal','dbase',3200);
INSERT INTO programmer VALUES('rebecca', '1967-01-01','1990-01-01','f','basic','cobol',2500);
INSERT INTO programmer VALUES('remitha ', '1970-04-19','1993-04-20','f','c','assembly',3600);
INSERT INTO programmer VALUES('revathi','1969-12-02','1992-01-02','f', 'pascal','basic',3700);
INSERT INTO programmer VALUES('vijaya','1965-12-14','1992-05-02','f','foxpro','c',3500);

INSERT INTO software VALUES('mary','readme','cpp',300.00,1200.00,84);
INSERT INTO software VALUES('anand', 'parachutes','basic', 399.95,6000.00, 43);
INSERT INTO software VALUES('anand', 'videotitling','pascal', 7500.00, 16000.00, 9);
INSERT INTO software VALUES('juliana', 'inventory','cobol', 3000.00, 3500.00, 0);
INSERT INTO software VALUES('kamala', 'payrollpkg','dbase', 9000.00, 20000.00, 7);
INSERT INTO software VALUES('mary', 'financialacct','oracle', 18000.00, 85000.00, 4);
INSERT INTO software VALUES('mary', 'codegenerator','c', 4500.00, 20000.00, 23);
INSERT INTO software VALUES('pattrick', 'readme','cpp', 300.00, 1200.00, 84);
INSERT INTO software VALUES('qadir', 'bombsaway','assembly', 750.00, 3000.00, 11);
INSERT INTO software VALUES('qadir', 'vaccines','c', 1900.00, 3100.00, 21);
INSERT INTO software VALUES('ramesh', 'hotelmgmt','dbase', 13000.00, 35000.00, 4);
INSERT INTO software VALUES('ramesh', 'deadlee','pascal', 599.95, 4500.00, 73);
INSERT INTO software VALUES('remitha ', 'pcutilities','c', 725.00, 5000.00, 51);
INSERT INTO software VALUES('remitha', 'tsrhelppkg','assembly', 2500.00, 6000.00, 7);
INSERT INTO software VALUES('revathi', 'hotelmgmt','pascal', 1100.00, 75000.00, 2);
INSERT INTO software VALUES('vijaya', 'tsreditor','c', 900.00, 700.00, 6);

INSERT INTO studies VALUES('anand', 'sabhari', 'pgdca', 4500);
INSERT INTO studies VALUES('altaf', 'coit', 'dca', 7200);
INSERT INTO studies VALUES('juliana', 'bdps', 'mca', 22000);
INSERT INTO studies VALUES('kamala', 'pragathi', 'dca', 5000);
INSERT INTO studies VALUES('mary', 'sabhari', 'pgdca', 4500);
INSERT INTO studies VALUES('nelson', 'pragathi', 'dap', 4500);
INSERT INTO studies VALUES('pattrick', 'pragathi', 'dcap', 6200);
INSERT INTO studies VALUES('qadir', 'apple', 'hdca', 14000);
INSERT INTO studies VALUES('ramesh', 'sabhari', 'pgdca', 4500);
INSERT INTO studies VALUES('rebecca', 'brilliant', 'dcap', 11000);
INSERT INTO studies VALUES('remitha ', 'bdps', 'dcs', 6000);
INSERT INTO studies VALUES('revathi', 'sabhari', 'dap', 5000);
INSERT INTO studies VALUES('vijaya', 'bdps', 'dca', 48000);
```

A Look at the Data: What We're Working With!

Before we start querying, let's see the actual data loaded into our tables. This gives you a clear picture of the information we'll be analyzing.

programmer Table Data

pname	dob	doj	gender	prof1	prof2	salary
Anand	1966-04-12	1992-04-21	m	pascal	basic	3200

pname	**dob**	**doj**	**gender**	**prof1**	**prof2**	**salary**
Altaf	1964-07-02	1990-11-13	M	Clipper	Cobol	2800
Juliana	1960-01-31	1990-04-21	F	Cobol	Dbase	3000
Kamala	1968-10-30	1992-01-02	F	C	Dbase	2900
Mary	1970-06-24	1991-02-01	F	Cpp	Oracle	4500
Nelson	1985-09-11	1989-10-11	M	Cobol	Dbase	2500
Pattrick	1965-11-10	1990-04-21	M	Pascal	Clipper	2800
Qadir	1965-08-31	1991-04-21	M	assembly	C	3000
Ramesh	1967-05-03	1991-02-28	M	Pascal	Dbase	3200
rebecca	1967-01-01	1990-01-01	f	Basic	Cobol	2500
remitha	1970-04-19	1993-04-20	f	c	assembly	3600
revathi	1969-12-02	1992-01-02	f	pascal	basic	3700
vijaya	1965-12-14	1992-05-02	f	foxpro	c	3500

software Table Data

pname	title	developin	scost	dcost	sold
mary	readme	cpp	300.00	1200.00	84
anand	parachutes	basic	399.95	6000.00	43
anand	videotitling	pascal	7500.00	16000.00	9
juliana	inventory	cobol	3000.00	3500.00	0
kamala	payrollpkg	dbase	9000.00	20000.00	7
mary	financialacct	oracle	18000.00	85000.00	4
mary	codegenerator	c	4500.00	20000.00	23
pattrick	readme	cpp	300.00	1200.00	84
qadir	bombsaway	assembly	750.00	3000.00	11
qadir	vaccines	c	1900.00	3100.00	21
ramesh	hotelmgmt	dbase	13000.00	35000.00	4
ramesh	deadlee	pascal	599.95	4500.00	73
remitha	pcutilities	c	725.00	5000.00	51
remitha	tsrhelppkg	assembly	2500.00	6000.00	7
revathi	hotelmgmt	pascal	1100.00	75000.00	2
vijaya	tsreditor	c	900.00	700.00	6

studies Table Data

pname	institute	course	coursefee
anand	sabhari	pgdca	4500
altaf	coit	dca	7200
juliana	bdps	mca	22000
kamala	pragathi	dca	5000
mary	sabhari	pgdca	4500
nelson	pragathi	dap	4500
pattrick	pragathi	dcap	6200
qadir	apple	hdca	14000
ramesh	sabhari	pgdca	4500
rebecca	brilliant	dcap	11000
remitha	bdps	dcs	6000
revathi	sabhari	dap	5000
vijaya	bdps	dca	48000

SQL Questions and Answers

1. **Find out the selling cost AVG for packages developed in Pascal.**

```
SELECT AVG(scost) FROM software WHERE developin = 'pascal';
```

2. **Display Names, Ages of all Programmers.**

```
SELECT
    pname,
    CAST(DATEDIFF(day, dob, GETDATE()) / 365.25 AS INT) AS age_in_years
FROM programmer;
```

---

3. **Display the Names and Date of Births of all Programmers Born in January.**

```
SELECT pname, dob FROM programmer WHERE MONTH(dob) = 1;
```

---

4. **How much revenue has been earned thru sales of Packages Developed in C.**

```
SELECT SUM(sold * scost) FROM software WHERE developin = 'c';
```

---

5. **Display details of Packages whose sales crossed the 2000 Mark.**

```
SELECT * FROM software WHERE (sold * scost) > 2000;
```

---

6. **Display the Details of Packages for which Development Cost have been recovered.**

```
SELECT * FROM software WHERE (sold * scost) >= dcost;
```

---

7. **How many Programmers Paid 5000 to 10000 for their course?**

```
SELECT COUNT(pname) FROM studies WHERE coursefee BETWEEN 5000 AND
10000;
```

---

8. **Display the details of the Programmers Knowing C.**

```
SELECT * FROM programmer WHERE prof1 = 'c' OR prof2 = 'c';
```

9. **How many Programmers know either COBOL or PASCAL.**

```
SELECT COUNT(pname) FROM programmer WHERE prof1 IN ('cobol', 'pascal') OR
prof2 IN ('cobol', 'pascal');
```

10. **How many Programmers Don't know PASCAL and C?**

```
SELECT COUNT(pname) FROM programmer
WHERE NOT (prof1 = 'pascal' OR prof2 = 'pascal' OR prof1 = 'c' OR prof2 = 'c');
```

11. **How old is the Oldest Male Programmer?**

```
SELECT TOP 1
    pname,
    CAST(DATEDIFF(day, dob, GETDATE()) / 365.25 AS INT) AS age_in_years
FROM programmer
WHERE gender = 'm'
ORDER BY age_in_years DESC;
```

12. **What is the AVG age of Female Programmers?**

```
SELECT AVG(
    CAST(DATEDIFF(day, dob, GETDATE()) / 365.25 AS INT)
) AS average_female_age
FROM programmer
WHERE gender = 'f';
```

13. **Calculate the Experience in Years for each Programmer and Display with their names in Descending order.**

```
SELECT
    pname,
    CAST(DATEDIFF(day, doj, GETDATE()) / 365.25 AS INT) AS experience_in_years
FROM programmer
ORDER BY experience_in_years DESC;
```

14. **Who are the Programmers who celebrate their Birthday's During the Current Month?**

```sql
SELECT pname, dob FROM programmer WHERE MONTH(dob) =
MONTH(GETDATE());
```

---

15. **What are the Languages studied by Male Programmers?**

```sql
SELECT DISTINCT prof1 AS language
        FROM programmer
        WHERE gender = 'm' AND prof1 IS NOT NULL
UNION
SELECT DISTINCT prof2
        FROM programmer
        WHERE gender = 'm' AND prof2 IS NOT NULL;
```

---

16. **Display the Cost of Package Developed By each Programmer.**

```sql
SELECT pname, SUM(scost) AS total_selling_cost FROM software GROUP BY pname;
```

---

17. **Display each language name with AVG Development Cost, AVG Selling Cost and AVG Price per Copy.**

```sql
SELECT
  developin AS language,
  AVG(dcost) AS avg_development_cost,
  AVG(scost) AS avg_selling_cost,
  AVG(scost) AS avg_price_per_copy
FROM software
GROUP BY developin;
```

---

18. **Display each programmer's name, costliest and cheapest Packages Developed by him or her.**

```sql
SELECT
  pname,
  MAX(dcost) AS costliest_package_dcost,
  MIN(dcost) AS cheapest_package_dcost
FROM software
GROUP BY pname;
```

---

19. **Display AVG Difference between SCOST, DCOST for Each Package.**

```
SELECT
   title AS package_title,
   AVG(scost - dcost) AS avg_difference
FROM software
GROUP BY title;
```

---

20. **Display the total SCOST, DCOST and amount to Be Recovered for each Programmer for Those Whose Cost has not yet been Recovered.**

```
SELECT
   pname,
   SUM(scost * sold) AS total_sales_value,
   SUM(dcost) AS total_development_cost,
   SUM(dcost - (sold * scost)) AS amount_to_be_recovered
FROM software
GROUP BY pname
HAVING SUM(dcost) > SUM(sold * scost);
```

---

21. **Who is the Highest Paid C Programmer?**

```
SELECT TOP 1 *
FROM programmer
WHERE (prof1 = 'c' OR prof2 = 'c')
ORDER BY salary DESC;
```

---

22. **Display the names of the highest paid programmers for each Language.**

```
WITH ProgrammerProficiencies AS (
   SELECT pname, salary, prof1 AS language FROM programmer WHERE prof1 IS NOT
NULL
   UNION ALL
   SELECT pname, salary, prof2 FROM programmer WHERE prof2 IS NOT NULL
),
RankedProficiencies AS (
   SELECT
      pname,
      salary,
      language,
      ROW_NUMBER() OVER (PARTITION BY language ORDER BY salary DESC) as
rn
   FROM ProgrammerProficiencies
)
SELECT pname, language, salary
FROM RankedProficiencies
WHERE rn = 1;
```

23. **Who is the least experienced Programmer?**

```
SELECT TOP 1
    pname,
    CAST(DATEDIFF(day, doj, GETDATE()) / 365.25 AS INT) AS experience_in_years
FROM programmer
ORDER BY experience_in_years ASC;
```

24. **Which Female Programmer earning more than 3000 does not know C, C++, ORACLE or DBASE?**

```
SELECT *
FROM programmer
WHERE gender = 'f'
  AND salary > 3000
  AND NOT (prof1 IN ('c', 'cpp', 'oracle', 'dbase') OR prof2 IN ('c', 'cpp', 'oracle', 'dbase'));
```

25. **What is the Costliest course?**

```
SELECT TOP 1 course, coursefee
FROM studies
ORDER BY coursefee DESC;
```

26. **Display the name of the Institute and Course, which has below AVG course fee.**

```
SELECT institute, course, coursefee
FROM studies
WHERE coursefee < (SELECT AVG(coursefee) FROM studies);
```

27. **Display the names of the courses whose fees are within 1000 (+ or -) of the Average Fee.**

```
SELECT course, coursefee
FROM studies
WHERE coursefee BETWEEN (SELECT AVG(coursefee) - 1000 FROM studies) AND
(SELECT AVG(coursefee) + 1000 FROM studies);
```

28. **Which course has below AVG number of Students?**

```
WITH CourseStudentCounts AS (
   SELECT course, COUNT(pname) AS student_count
   FROM studies
   GROUP BY course
)
SELECT course, student_count
FROM CourseStudentCounts
WHERE student_count < (SELECT AVG(student_count) FROM CourseStudentCounts);
```

---

29. **Which language was used to develop the most number of Packages?**

```
SELECT TOP 1 developin, COUNT(title) AS package_count
FROM software
GROUP BY developin
ORDER BY package_count DESC;
```

---

30. **Which programmer has developed the highest number of Packages?**

```
SELECT TOP 1 pname, COUNT(title) AS package_count
FROM software
GROUP BY pname
ORDER BY package_count DESC;
```

---

31. **Who are the authors of the Packages, which have recovered more than double the Development cost?**

```
SELECT DISTINCT pname
FROM software
WHERE (sold * scost) > (2 * dcost);
```

---

32. **Display the programmer Name and the cheapest packages developed by them in each language.**

```
WITH RankedPackages AS (
   SELECT
      pname,
      title,
      developin,
      dcost,
        ROW_NUMBER() OVER (PARTITION BY developin ORDER BY dcost ASC) as rn
    FROM software
)
SELECT pname, title, developin, dcost
FROM RankedPackages
WHERE rn = 1;
```

---

33. **Display the language used by each programmer to develop the Highest Selling and Lowest-selling package.**

```
WITH RankedSales AS (
    SELECT
        pname,
        developin,
        (sold * scost) AS sales_value,
        ROW_NUMBER() OVER (PARTITION BY pname ORDER BY (sold * scost) DESC)
as rn_highest,
        ROW_NUMBER() OVER (PARTITION BY pname ORDER BY (sold * scost) ASC)
as rn_lowest
    FROM software
)
SELECT pname, developin AS highest_selling_language FROM RankedSales WHERE
rn_highest = 1
UNION
SELECT pname, developin AS lowest_selling_language FROM RankedSales WHERE
rn_lowest = 1;
```

---

34. **Who is the youngest male Programmer born in 1965?**

```
SELECT TOP 1 pname, dob
FROM programmer
WHERE gender = 'm' AND YEAR(dob) = 1965
ORDER BY dob DESC;
```

---

35. **In which year was the most number of Programmers born?**

```
SELECT TOP 1 YEAR(dob) AS birth_year, COUNT(pname) AS programmer_count
FROM programmer
GROUP BY YEAR(dob)
ORDER BY programmer_count DESC;
```

---

36. **In which month did most number of programmers join?**

```
SELECT TOP 1 MONTH(doj) AS join_month, COUNT(pname) AS programmer_count
```

```
FROM programmer
GROUP BY MONTH(doj)
ORDER BY programmer_count DESC;
```

---

37. **In which language are most of the programmer's proficient?**

```
WITH AllProficiencies AS (
  SELECT prof1 AS language FROM programmer WHERE prof1 IS NOT NULL
  UNION ALL
  SELECT prof2 FROM programmer WHERE prof2 IS NOT NULL
)
SELECT TOP 1 language, COUNT(language) AS proficiency_count
FROM AllProficiencies
GROUP BY language
ORDER BY proficiency_count DESC;
```

---

38. **Who are the male programmers earning below the AVG salary of Female Programmers?**

```
SELECT pname, salary
FROM programmer
WHERE gender = 'm' AND salary < (SELECT AVG(salary) FROM programmer WHERE
gender = 'f');
```

---

39. **Who are the Female Programmers earning more than the Highest Paid Male Programmer?**

```
SELECT pname, salary
FROM programmer
WHERE gender = 'f' AND salary > (SELECT MAX(salary) FROM programmer WHERE
gender = 'm');
```

---

40. **Display the details of those who are drawing the same salary.**

```
SELECT p1.*
FROM programmer p1
JOIN programmer p2 ON p1.salary = p2.salary AND p1.pname <> p2.pname
ORDER BY p1.salary, p1.pname;
```

---

41. **Display the details of the Software Developed by the Male Programmers Earning More than 3000/-.**

```
SELECT s.*
FROM software s
JOIN programmer p ON s.pname = p.pname
WHERE p.gender = 'm' AND p.salary > 3000;
```

---

42. **Display the details of the Software Developed in C By female programmers of Pragathi.**

```
SELECT s.*
FROM software s
JOIN programmer p ON s.pname = p.pname
JOIN studies st ON p.pname = st.pname
WHERE s.developin = 'c' AND p.gender = 'f' AND st.institute = 'pragathi';
```

---

43. **Display the details of the software Developed in DBASE by Male Programmers, who belong to the institute in which most number of Programmers studied.**

```
SELECT s.*
FROM software s
JOIN programmer p ON s.pname = p.pname
JOIN studies st ON p.pname = st.pname
WHERE s.developin = 'dbase'
  AND p.gender = 'm'
  AND st.institute = (
      SELECT TOP 1 institute
      FROM studies
      GROUP BY institute
      ORDER BY COUNT(pname) DESC
  );
```

---

44. **Display the details of the software that has developed in the language which is neither the first nor the second proficiency of the programmers.**

```
SELECT s.*
FROM software s
JOIN programmer p ON s.pname = p.pname
WHERE s.developin <> p.prof1 AND s.developin <> p.prof2;
```

---

45. **Display the names of the programmers who have not developed any packages.**

```
SELECT pname
FROM programmer
WHERE pname NOT IN (SELECT DISTINCT pname FROM software);
```

---

46. **Who are the programmers who joined on the same day?**

```
SELECT p1.pname, p1.doj
FROM programmer p1
JOIN programmer p2 ON p1.doj = p2.doj AND p1.pname <> p2.pname
ORDER BY p1.doj, p1.pname;
```

---

47. **How many packages were developed by students, who studied in institute that charge the lowest course fee?**

```
SELECT COUNT(s.title) AS number_of_packages
FROM software s
JOIN studies st ON s.pname = st.pname
WHERE st.coursefee = (SELECT MIN(coursefee) FROM studies);
```

---

48. **How many packages were developed by the female programmers earning more than the highest paid male programmer?**

```
SELECT COUNT(DISTINCT s.title)
FROM software s
JOIN programmer p ON s.pname = p.pname
WHERE p.gender = 'f' AND p.salary > (SELECT MAX(salary) FROM programmer
WHERE gender = 'm');
```

---

49. **How many packages are developed by the most experienced programmer from BDPS?**

```
SELECT COUNT(s.title) AS number_of_packages
FROM software s
JOIN programmer p ON s.pname = p.pname
JOIN studies st ON p.pname = st.pname
WHERE st.institute = 'bdps'
ORDER BY DATEDIFF(day, p.doj, GETDATE()) DESC
LIMIT 1;
```

---

50. **List each PROF with the number of Programmers having that PROF and the number of the packages in that PROF.**

```
WITH AllProficiencies AS (
    SELECT prof1 AS proficiency FROM programmer WHERE prof1 IS NOT NULL
    UNION ALL
    SELECT prof2 FROM programmer WHERE prof2 IS NOT NULL
)
SELECT
    ap.proficiency,
    COUNT(DISTINCT p.pname) AS number_of_programmers,
    COUNT(s.title) AS number_of_packages_developed_in_prof
FROM AllProficiencies ap
LEFT JOIN programmer p ON ap.proficiency = p.prof1 OR ap.proficiency = p.prof2
LEFT JOIN software s ON ap.proficiency = s.developin
GROUP BY ap.proficiency;
```

---

51. **Calculate the total number of students for each course, but only for courses with more than 2 students.**

```
SELECT course, COUNT(pname) AS number_of_students
FROM studies
GROUP BY course
HAVING COUNT(pname) > 2
ORDER BY number_of_students DESC;
```

---

52. **For each institute, find the most expensive course offered.**

```
SELECT s1.institute, s1.course, s1.coursefee
FROM studies s1
JOIN (
    SELECT institute, MAX(coursefee) AS max_fee
    FROM studies
    GROUP BY institute
) s2 ON s1.institute = s2.institute AND s1.coursefee = s2.max_fee;
```

---

53. **List programmers who know 'C' as their primary proficiency (prof1) but haven't developed any software in 'C'.**

```
SELECT p.pname, p.prof1
FROM programmer p
LEFT JOIN software s ON p.pname = s.pname AND s.developin = 'c'
WHERE p.prof1 = 'c' AND s.title IS NULL;
```

---

54. **Which programmer has the most diverse skill set (i.e., knows the most different programming languages)?**

```
SELECT TOP 1 pname, COUNT(DISTINCT prof) AS skill_count
FROM (
   SELECT pname, prof1 AS prof FROM programmer WHERE prof1 IS NOT NULL
   UNION ALL
   SELECT pname, prof2 AS prof FROM programmer WHERE prof2 IS NOT NULL
) AS combined
GROUP BY pname
ORDER BY skill_count DESC;
```