

Z

Q4

```
SELECT Student_id,
       name,
       age,
       department
  FROM Students
 WHERE age BETWEEN 21 AND 23;
```

Outputs

Student_id	name	age	department
2	Bob	22	HR
3	Charlie	21	IT
4	Diana	23	Finance
5	Eve	22	HR

Exercise 2 : SQL Aggregate Functions  
 & SQL Operators

Q1

```
SELECT DISTINCT department
  FROM Students;
```

Output

## Department

IT

HR

Finance

Q2

```
SELECT department,
       AVG(age) AS avg-age
  FROM Students
 GROUP BY department;
```

Output

## department | Avg-age

IT | 20.5

HR | 22.0

Finance | 23.0

Q5

```
SELECT Student_id, name, age, department
  FROM Students
 WHERE (department IN ('IT', 'HR'))
   AND age > 21;
```

Outputs

Student_id	name	age	department
2	Bob	22	HR
5	Eve	22	HR

Q3

```
SELECT department,
       COUNT(*) AS student_count
  FROM Students
 GROUP BY department
 HAVING COUNT(*) > 1;
```

Q6

```
SELECT department, SUM(credits)
       AS total_credits
  FROM Courses
 GROUP BY department
 HAVING SUM(credits) > 5;
```

Output

department	student_count
IT	2
HR	2

Outputs

department	Total_credits
IT	11

Q6

```
SELECT course_id
    COUNT(*) AS enrollment_count
  FROM enrollments
 GROUP BY course_id;
```

course_id	enrollment_count
101	1
102	1
103	1
104	1
105	1

Q7

```
SELECT course_id, course_name,
       department, credits
  FROM courses
 WHERE credits < 4;
```

course_id	course_name	department	credits
101	SQL Basics	IT	3
104	Excel	Finance	2
105	Statistics	HR	3

Q8

```
SELECT department,
       SUM(salary) AS total_salary,
       SUM(bonus) AS total_bonus
  FROM Salaries
 GROUP BY department;
```

department	Total salary	Total bonus
IT	122 000	10500
HR	109 000	7500
Finance	70 000	6000

Q8

```
SELECT course_id, course_name, credit
  FROM courses
 ORDER BY credit DESC
 LIMIT 3;
```

course_id	course_name	credits
102	Python	4
103	Data Science	4
101	SQL Basics	3

Q12

```
SELECT department,
       AVG(salary) AS avg_salary
  FROM Salaries
 GROUP BY department
 HAVING AVG(salary) > 55000;
```

department	avg_salary
IT	61 000
Finance	70 000

Q9

```
SELECT MAX(grade) AS max_grade,
       MIN(grade) AS min_grade,
       AVG(grade) AS avg_grade
  FROM enrollments;
```

Max-grade	Min-grade	avg-grade
90	78	84.6

Q13 SELECT project-id, project-name, department, budget  
 FROM projects  
 WHERE budget BETWEEN 50 000 AND 120 000  
 AND department <> 'Marketing';

Output

Project-id	Project-name	Department	Budget
1	AI App	IT	120 000
2	Payroll system	Finance	80 000
5	HR Portal	HR	50 000

Q13 SELECT employee-id, name, salary, bonus,  
 $(\text{salary} + \text{bonus})$  AS total\_compo  
 FROM salaries  
 WHERE  $(\text{salary} + \text{bonus}) > 60 000$ ;

Output

emp-id	name	salary	bonus	Total Compo
1	Tom	60 000	5000	65000
3	Spike	70 000	6000	76000
4	Tyke	62000	5500	67500

Q14 SELECT department  
 $\text{sum}(\text{budget})$  AS total-budget,  
 $\text{avg}(\text{budget})$  AS avg-budget  
 FROM projects  
 GROUP BY department  
 HAVING avg(budget) > 2000;

Output

Department	Total Budget	Avg-budget
IT	270 000	135000
Finance	80 000	80 000