BRIGHTLEARN TUTORIALS

DATA ANALYTICS

EXERCISE 2 : SQL AGGREGATE FUNCTIONS & SQL OPERATORS

Table : students

1. List all distinct department

Syntax :

```
SELECT DISTINCT department
FROM students;
```

Output :

department
IT
HR
Finance

2. Average age of students per department

Syntax :

```
SELECT department,
AVG (age) AS average_age
FROM students
GROUP BY department;
```

Output :

department	average_age
IT	20.5
HR	22
Finance	23

3. Department with more than 1 student

Syntax :

```
SELECT department,
COUNT (*) AS Student-count
FROM students
GROUP BY department
HAVING COUNT(*) >1 ;
```

Output :

department	Student_count
IT	2
HR	2

4. Student whose age is between 21 and 23

Syntax :

```
SELECT student_id,
       name,
       age,
       department
FROM students
WHERE age BETWEEN 21 AND 23;
```

Output

Student-id	name	age	department
2	Bob	22	HR
5	Eve	22	HR

5. List all students in the IT or HR department who are older than 21

Syntax :

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

Output :

student_id	name	age	department
2	Bob	22	HR
~~4~~	~~Diana~~	~~23~~	
5	Eve	22	HR

Table : courses

6. Total credits per department, only for departments with more than 5 total credits

Syntax :

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

Output :

department	total_credits
IT	11

7. List all courses that do not have 4 credits

Syntax :

```
SELECT course_id,
       course_name,
       department,
       Credits
FROM courses
WHERE credit != 4;
```

Output :

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

8. Top 3 courses by credits descending

Syntax :

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

Output :

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

Table : enrollments

9. Maximum, minimum and average grades

Syntax:

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

Output:

Max_grade	min_grade	avg_grade
90	78	84.6

10. How many enrollments exist per course

Syntax:

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

Output:

Course_id	enrollment_count
101	1
102	1
103	1
104	1
105	1

Table: salaries

11. Total salary and bonus per department

Syntax :

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

Output :

department	total_salary	total_bonus
IT	122000	10500
HR	109000	7500
Finance	70000	6000

12. Departments with average salary above 55000

Syntax :

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

Output :

department	avg_salary
IT	61000
Finance	70000

13. Employees whose salary plus bonus is more than 60000

Syntax :

```
SELECT employee_id,
       name,
       salary,
       bonus,
(salary + bonus) AS total_compensation
FROM salaries
WHERE (salary + bonus) > 60000;
```

Output :

employee_id	name	salary	bonus	total_compensation
1	Tom	60000	5000	65000
3	Spike	70000	6000	76000
4	Tyke	62000	5500	67500

Table : Projects

14. Total and average budget per department with average budget above 70000

Syntax :

```
SELECT department,
       SUM (budget) AS total_budget,
       AVG (budget) AS avg_budget
FROM projects
GROUP BY department
HAVING AVG (budget) > 70000;
```

Output :

department	total_budget	avg_budget
IT	270000	135000
Finance	80000	80000

15. All projects with budget between 50000 and 120000 excluding Marketing

Syntax :

```
SELECT project_id,
       project_name,
       department,
       budget
FROM projects
WHERE budget BETWEEN 50000 AND 120000
    AND department != 'Marketing';
```

Output :

project_id	project_name	department	budget
2	Payroll System	Finance	80000