Aggregate Functions

Aggregate Function:

Aggregate functions are functions that take a collection of values as input and return a single value.

- Behavior of Aggregate Functions:
  - Operates - on a single column
  - Return - a single value.
  - Used only in the SELECT list and in the HAVING clause.

Behavior of Aggregate function Continued…

- Accepts:

  ✔ DISTINCT : consider only distinct values of the argument expression.

  ✔ ALL : consider all values including all duplicates.

  Example: SELECT COUNT(DISTINCT column_name)

Types of SQL Aggregate Functions

- SUM
- AVG
- MIN
- MAX
- COUNT

Input to Aggregate Function

- SUM and AVG :
  - Operates only on collections of numbers .

- MIN , MAX and COUNT
  - Operates on collection of numeric and non-numeric data types.

❑ Each function eliminates NULL values and operates on Non- null values.

Staff

sno	fname	lname	salary	position
SL100	John	White	30000.00	Manager
SL101	Susan	Brand	24000.00	Manager
SL102	David	Ford	12000.00	Project Manager
SL103	Ann	Beech	12000.00	Project Manager
SL104	Mary	Howe	9000.00	Project Manager

SUM()

Returns: The sum of the values in a specified column.

Example: Find the total/sum of the Managers salary

Query:

```
SELECT  SUM( salary) AS sum_salary
FROM Staff
WHERE position = 'Manager';
```

Result:

sum_salary
54000.00

AVG()

Returns: The average of the values in a specified column.

Example: Find the average of the Project Managers salary .

Query:

```
SELECT  AVG(  salary) AS avg_salary
FROM  Staff
WHERE position = 'Project Manager';
```

Result:

avg_salary
10500.00

// Error in Result

// avg_salary = 11000.00

// What is wrong?

Revised Query for AVG()

Query:

```
SELECT  AVG(ALL salary) AS avg_salary
FROM  Staff
WHERE position = 'Project Manager';
```

Result :

	avg_salary
1	1000.00

CAUTION: Using DISTINCT and ALL in SUM() and AVG()

Staff

sno	fname	lname	salary	position
SL100	John	White	30000.00	Manager
SL101	Susan	Brand	24000.00	Manager
SL102	David	Ford	12000.00	Project Manager
SL103	Ann	Beech	12000.00	Project Manager
SL104	Mary	Howe	9000.00	Project Manager

MIN() and MAX()

Returns: MIN() returns the smallest value of a column.
MAX() returns the largest value of a column.

Example: Find the minimum and maximum staff salary.

Query:

```
SELECT  MIN( salary) AS min_salary, MAX (salary)  AS max_salary
FROM  Staff;
```

Result:

min_salary	max_salary
9000.00	30000.00

COUNT()

Returns: The number of values in the specified column.

Example: Count number of staffs who are Manager.

Query: SELECT COUNT(sno) AS sno_count
FROM Staff
WHERE Staff.position = 'Manager';

Result:

sno_count
2

Use of COUNT() and SUM()

Example: Find the total number of Managers and the sum of there salary.

Query: SELECT COUNT(sno) AS sno_count , SUM(salary) AS sum_salary
From Staff
WHERE position = 'Manager';

sno	fname	lname	salary	position
SL100	John	White	30000.00	Manager
SL101	Susan	Brand	24000.00	Manager

COUNT

SUM

COUNT() and SUM() continued

- Result:

sno_count	sum_salary
2	54000.00

Staff

sno	fname	lname	salary	position
SL100	John	White	30000.00	Manager
SL101	Susan	Brand	24000.00	Manager
SL102	David	Ford	12000.00	Project Manager
SL103	Ann	Beech	12000.00	Project Manager
SL104	Mary	Howe	9000.00	Project Manager

COUNT(*)

Input: There is no input to this function.

Returns: It counts all the rows of a table , regardless of whether Nulls or the duplicate occur.

Example: How many Project Manager salary is more than 9000.00

Query: SELECT COUNT(*) AS Count_Salary
FROM Staff
WHERE position = 'Project Manager' AND
Staff.salary > 9000.00

COUNT(*) continued…

- Result:

Count_ Salary
2

Usage of Aggregation Functions

- Use of GROUP BY
- Use of HAVING

Use of GROUP BY

- GROUP BY: It groups the data from the SELECT table and produce a single summary row for each group

- When Using GROUP BY:

1. Each item in the SELECT list must be single valued per group.
2. SELECT clause may only contain:
   - Columns names
   - Aggregate function
   - Constants
   - An expression involving combinations of the above.

3. All column names in SELECT list must appear in the GROUP BY clause unless the name is used only in the aggregate function.

Staff

sno	bno	fname	lname	salary	position
SL100	B3	John	White	30000.00	Manager
SL101	B5	Susan	Brand	24000.00	Manager
SL102	B3	David	Ford	12000.00	Project Manager
SL103	B5	Ann	Beech	12000.00	Project Manager
SL104	B7	Mary	Howe	9000.00	Project Manager

GROUP BY

Example: Find the number of staff working in each branch and the sum of their salaries.

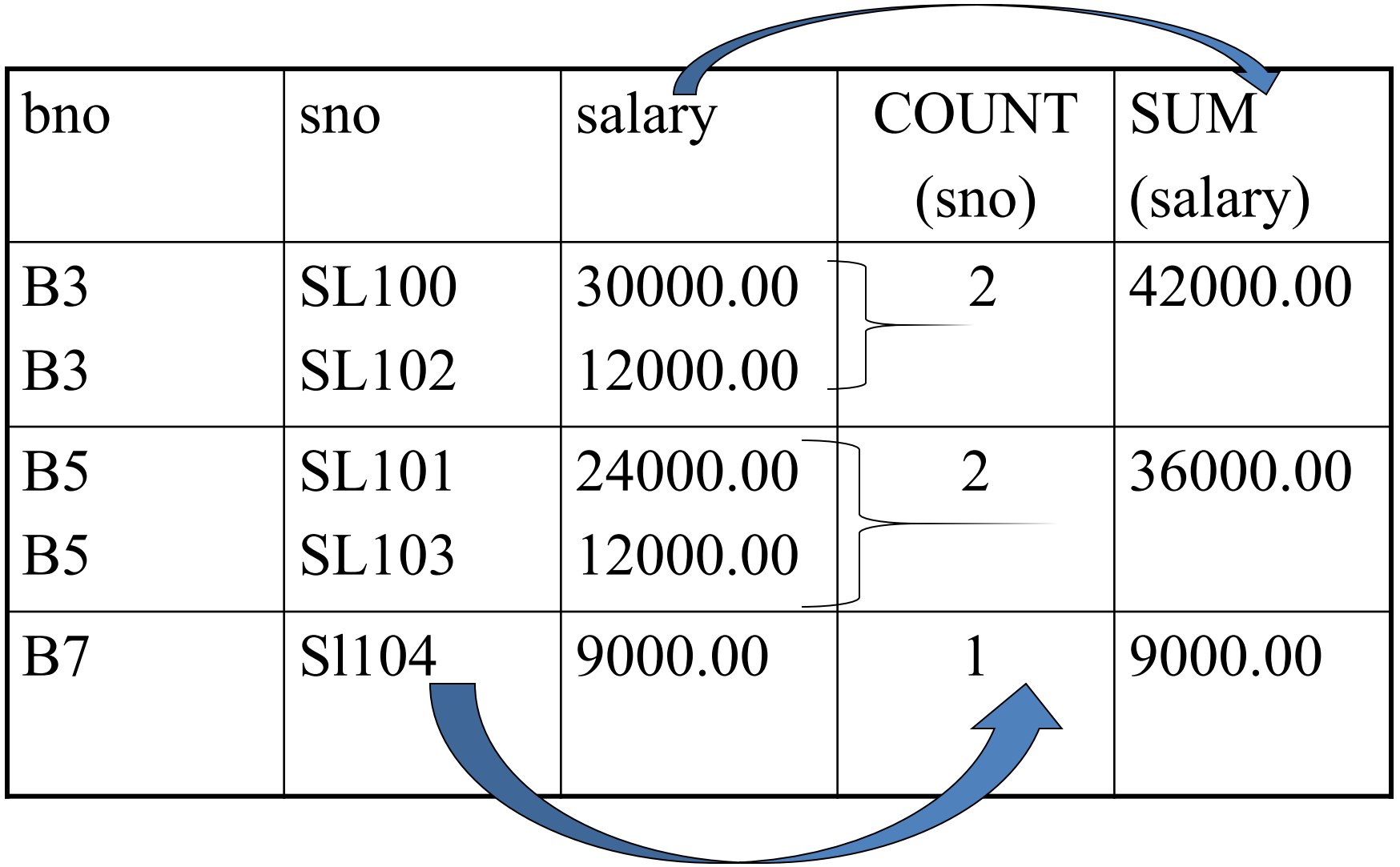
Query:

```
SELECT bno, COUNT(sno) AS count, SUM(salary) AS sum
FROM Staff
GROUP BY bno
ORDER by bno;
```

Result:

bno	count	sum
B3	2	42000.00
B5	2	36000.00
B7	1	9000.00

SQL'S ROLE



bno	sno	salary	COUNT (sno)	SUM (salary)
B3	SL100	30000.00	2	42000.00
B3	SL102	12000.00		
B5	SL101	24000.00	2	36000.00
B5	SL103	12000.00		
B7	Sl104	9000.00	1	9000.00

USE OF HAVING

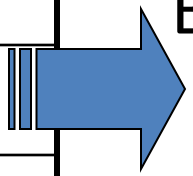
HAVING clause: It is designed to be used with GROUP BY so that it can restrict the groups that appear in the final result table.

Example: For each branch office with more than one member of staff, find the number of staff working in each branch and the sum of their salaries.

Query: SELECT bno, COUNT(sno) AS count, SUM(salary) AS sum
FROM Staff
GROUP BY bno
HAVING COUNT(sno) > 1
ORDER by bno;

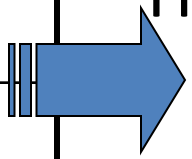
Having Clause continued….

bno	COUNT (sno)	SUM (salary)
B3	2	42000
B5	2	36000
B7	1	9000



- Result table after performing GROUP BY bno clause.

bno	count	sum
B3	2	42000
B5	2	36000



- Final result table after performing HAVING COUNT(sno) 1

  ORDER by bno;