Data Manupilation Language (DML)- Group and Aggregate Functions

P.Mirunalini

SSNCE

March 15, 2022

Session Objective

- To learn about the DML statements
- To learn about Aggregate and group functions

Table of Contents

SQL Functions

Functions are a very powerful feature of SQL and can be used to do the following:

- Perform calculations on data
- Modify individual data items
- Manipulate output for groups of rows
- Format dates and numbers for display

SQL functions sometimes take arguments and always return a value.

Employee Database

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
100	Steven	King	SKING	515.123.4567	17-JUN-87	AD_PRES	24000	-	-	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	17000	-	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	17000	-	100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-90	IT_PROG	9000	-	102	60
104	Bruce	Ernst	BERNST	590.423.4568	21-MAY-91	IT_PROG	6000	-	103	60
105	David	Austin	DAUSTIN	590.423.4569	25-JUN-97	IT_PROG	4800	-	103	60
106	Valli	Pataballa	VPATABAL	590.423.4560	05-FEB-98	IT_PROG	4800	-	103	60
107	Diana	Lorentz	DLORENTZ	590.423.5567	07-FEB-99	IT PROG	4200	-	103	60
124	Kevin	Mourgos	KMOURGOS	650.123.5234	16-NOV-99	ST_MAN	5800	-	100	50
141	Trenna	Rajs	TRAJS	650.121.8009	17-OCT-95	ST_CLERK	3500	-	124	50
142	Curtis	Davies	CDAVIES	650.121.2994	29-JAN-97	ST_CLERK	3100	-	124	50
143	Randall	Matos	RMATOS	650.121.2874	15-MAR-98	ST_CLERK	2600	-	124	50
144	Peter	Vargas	PVARGAS	650.121.2004	09-JUL-98	ST_CLERK	2500	-	124	50
149	Eleni	Zlotkey	EZLOTKEY	011.44.1344.429018	29-JAN-00	SA_MAN	10500	.2	100	80
174	Ellen	Abel	EABEL	011.44.1644.429267	11-MAY-96	SA_REP	11000	.3	149	80
176	Jonathon	Taylor	JTAYLOR	011.44.1644.429265	24-MAR-98	SA_REP	8600	.2	149	80
178	Kimberely	Grant	KGRANT	011.44.1644.429263	24-MAY-99	SA_REP	7000	.15	149	-
200	Jennifer	Whalen	JWHALEN	515.123.4444	17-SEP-87	AD_ASST	4400	-	101	10
201	Michael	Hartstein	MHARTSTE	515.123.5555	17-FEB-96	MK_MAN	13000	-	100	20
202	Pat	Fay	PFAY	603.123.6666	17-AUG-97	MK_REP	6000	-	201	20
205	Shelley	Higgins	SHIGGINS	515.123.8080	07-JUN-94	AC_MGR	12000	-	101	110
206	William	Gietz	WGIETZ	515.123.8181	07-JUN-94	AC_ACCOUNT	8300	-	205	110

Types of SQL Functions

There are two distinct types of functions:

- Single-row functions
- Multiple-row functions

1. Single-Row Functions: These functions operate on single rows only and return one result per row.
2. Multiple-Row Functions : Functions can manipulate groups of rows to give one result per group of rows.
   These functions are known as group functions.

Single-Row Functions

- Manipulate data items
- Act on each row returned in the query
- Return one result per row
- Can be used in SELECT, WHERE, and ORDER BY clauses; can be nested

Single-Row Functions

- **Character functions**: Accept character input and can return both character and number values
- **Number functions**: Accept numeric input and return numeric values
- **Date functions**: Operate on values of the DATE data type
- **Conversion functions**: Convert a value from one data type to another

Character Functions- Case-manipulation functions

Character functions can be divided into the following:

- Case-manipulation functions
- Character-manipulation functions

Case-manipulation functions

1. **LOWER(column /expression)**: Converts alpha character values to lowercase
2. **UPPER(column/ expression)** : Converts alpha character values to uppercase
3. **INITCAP(column/expression)** : Converts alpha character values to uppercase for the first letter of each word, all other letters in lowercase

Frame Title

- **CONCAT(column1 | expression1,column2 |expression2)** : Concatenates the first character value to the second character value; equivalent to concatenation operator (||)
- **SUBSTR(column|expression,m)** : Returns specified characters from character value starting at character position m.
- **LENGTH(column|expression)**; Returns the number of characters in the expression.
- **INSTR(column—expression,'string')**; Returns the numeric position of a named string.

Character-manipulation functions

- **LPAD(column |expression, n, 'string')** : Pads the character value right-justified to a total width of n character positions.
- **RPAD(column|expression, n, 'string')** : Pads the character value left-justified to a total width of n character positions
- **TRIM(leading|trailing|both)** : Enables to trim heading or trailing characters (or both) from a character string
- **REPLACE(text, search_string, replacement_string)**: Searches a text expression for a character string and, if found, replaces it with a specified replacement string.

Singe row functions- example

```
1.select last_name,lower(last_name)  Name,
initcap(lower(last_name))  int_Name,
upper(first_name)  u_Name from employees;
```

last_Name	Name	int_Name	u_Name
Abel	abel	Abel	ELLEN
Austin	austin	Austin	DAVID
Davies	davies	Davies	CURTIS

Singe row functions- example

```
2. select last_name,substr(last_name,1,5) substr from
 employees;
```

last_Name	substr
Austin	Austi
Davies	Davie

```
3. select last_name,substr(last_name,-1) substr from
 employees;
```

Last_Name	substr
Abel	l
Austin	n

Singe row functions- example

4. `select first_name, length(first_name) length from employees;`

First_Name	length
Ellen	5
David	5

5. `select employee_id,last_name,instr(last_name,'a',1,2) from employees;`

lemployee_id	last_name	instr(last_name,'a',1,2)
142	Davies	0
102	De Haan	6
106	Pataballa	4

Singe row functions- example

6. `select rpad(salary,10,'*'), lpad(salary,10,'*') from employees;`

RPAD(SALARY)	LPAD(SALARY)
24000*****	*****24000

7. `select trim('H' from 'HELLOWORLD') from dual;`

TRIM('H'F)
ELLOWORLD

8. `select round(43.235,2) r,trunc(43.235,1) t, ceil(41.235) c,floor(41.235) f from dual;`

r	t	c	f
43.24	43.2	42	41

Date Functions

- MONTHS_BETWEEN Number of months between two dates
- ADD_MONTHS Add calendar months to date
- NEXT_DAY Next day of the date specified
- LAST_DAY Last day of the month
- ROUND Round date
- TRUNC Truncate date

Date Functions

```
9. select sysdate from dual;
 Sysdate
 08-FEB-21
```

```
10. select sysdate,sysdate-5 from dual;
```

08-FEB-21, 03-feb-21

```
11. select hire_date,  months_between(sysdate,hire_date)
mon  from employees;
```

HIRE_DATE	mon
17-JUN-87	319.130284

```
12. select last_name,(sysdate-hire_date)/7 as weeks
 from employees where department_id=90
```

LAST_NAME	WEEKS
King	1387.86284
Kochhar	1269.71998

Date Functions

```
13. Select employee_id id ,hire_date h_d,
months_between(sysdate,hire_date) tenure,
add_months(hire_date,6) review
from employees where months_between(sysdate,hire_date)>3
```

id	hd	tenure	review
100	17-JUN-87	19.130992	17-DEC-87
101	21-SEP-89	292	21-MAR-90

```
14. select next_day(sysdate,'FRIDAY') from dual;
```

NEXT_DAY(10-JAN-20)

```
15. select to_char(sysdate,'ddspth mon yyyy') from dual;
TO_CHAR(SYSDATE,'DDSPTH')
```

Eigth Feb 2021

Group Functions

```
16.select avg(salary) avg,max(salary) max,min(salary) min,
sum(salary) sum from employees where job_id like 'SA\_Rep';
```

avg	max	min	sum
8150	11000	6000	32600

```
17. select min(hire_date) min,max(hire_date) max
 from employees;
```

MIN	MAX
17-JUN-87	29-JAN-00

```
18. select min(last_name),max(last_name)
from employees
```

MIN(LAST_NAME)	MAX(LAST_NAME)
Abel	Zlotkey

Group Functions

All columns in the SELECT list that are not in group functions must be in the GROUP BY clause.

- The SELECT clause specifies the columns to be retrieved:
- The FROM clause specifies the tables that the database must access
- The GROUP BY clause specifies how the rows should be grouped.
- Any column or expression in the SELECT list that is not an aggregate function must be in the GROUP BY clause

```
18. SELECT department_id, AVG(salary) FROM
employees GROUP BY department_id;
```

The GROUP BY column does not have to be in the SELECT list.

```
19. SELECT AVG(salary) FROM employees GROUP BY department_id;
```

GROUP BY Clause on Multiple Columns

- Return summary results for groups and subgroups by listing more than one GROUP BY column.
- Determine the default sort order of the results by the order of the columns in the GROUP BY clause

```
20. SELECT department_id dept_id, job_id, SUM(salary)
FROM employees GROUP BY department_id, job_id;
```

DEPT_ID	JOB_ID	SUM(SALARY)
10	AD_ASST	4400
20	MK_MAN	13000
20	MK_REP	6000
50	ST_CLERK	11700
50	ST_MAN	5800
60	IT_PROG	19200
80	SA_MAN	10500
80	SA_REP	19600
90	AD_PRES	24000
90	AD_VP	34000

Group functions- example

- Cannot use the WHERE clause to restrict groups.
- Use the HAVING clause to restrict groups.
- Cannot use group functions in the WHERE clause.

```
SELECT department_id, AVG(salary)
FROM employees
HAVING AVG(salary) > 8000
GROUP BY department_id;
```

Group functions - example

- To list the sum of the salary earned and their department number by the employees who are working under manager =100
- List the no of employees their department_code under each manager of the departments.
- List the department whose average salary greater than 10,000 .
- List th employees whose sum_salary >13000 and job_id is not to be “REP”
- List th employees whose sum_salary >13000 and job_id is not to be “REP” and order it based on sum(salary)
- Display the data for those employees whose last names end with an n or job_id is ‘rep’

Group functions - Queries

```
21. select department_id,manager_id,sum(salary) from employees
group by department_id,manager_id having manager_id=100;

22. SELECT department_id "Department Code", manager_id,
count(employee_id )  FROM  employees
group by department_id, manager_id;

 23. SELECT department_id "Department Code",
  round(avg(salary)) "Average Salary"  FROM  employees
  GROUP BY department_id  having avg(salary)>10000;
```

Group functions - Queries

```
24. select job_id,sum(salary) payroll from employees
group by job_id
having sum(salary)>13000
and job_id not like '%REP%'

25. select job_id,sum(salary) payroll from employees
group by job_id
 having sum(salary)>13000
  and job_id not like '%REP%'
 order by sum(salary);

26. SELECT employee_id, CONCAT(first_name, last_name) NAME,
LENGTH (last_name), INSTR(last_name, 'a') "Contains 'a'?"
FROM employees WHERE SUBSTR(job_id, 4) = 'REP';
```

Reference

- Fundamentals of Database systems 7^{th} Edition by Ramez Elmasri.
- Oracle 9i SQL