

Chapter 6 Aggregate Functions, Group By, and Having Clauses

Aggregate Functions



- Are multi-row functions
- Work on groups of rows and return a single result for each group

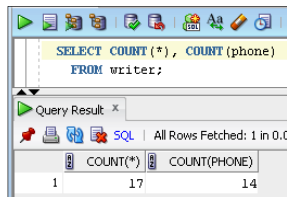
Function	Description
COUNT(*)	returns the number of rows, including nulls and duplicates
COUNT((DISTINCT ALL) expr)	returns the number of nonnull rows
SUM((DISTINCT ALL) expr)	returns numerical sum; ignores nulls
AVG((DISTINCT ALL) expr)	returns the numerical average; ignores nulls
MAX((DISTINCT ALL) expr)	returns largest value; ignores nulls
MIN((DISTINCT ALL) expr)	returns smallest value; ignores nulls

2

Aggregate Functions and NULLs

- Aggregate functions ignore Nulls

- except COUNT(*)

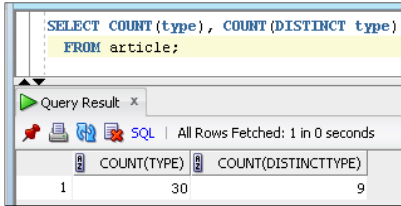


- Can use a function to provide a substitute value for a Null
 - eg: NVL, COALESCE

3

COUNT and DISTINCT

- Returns the number of distinct values



```
SELECT COUNT(type), COUNT(DISTINCT type)
FROM article;
```

Query Result x

All Rows Fetched: 1 in 0 seconds

	COUNT(TYPE)	COUNT(DISTINCTTYPE)
1	30	9

4

Aggregate Function Practice

- How many records are in the writer table?
- How many writers have a phone number listed?
- How many writers don't have a phone number listed?
- What is the total length and average length of all articles? Also show the length of both the longest and shortest article.
- Modify the previous query to show the same summary results for only business and law articles.

5

GROUP BY Clause

- Divides rows into subgroups
- When used with aggregate functions, the query returns one row of aggregate results for each group
- Column aliases are not allowed in GROUP BY clause
- Use an ORDER BY clause to control row sort order
 - <Oracle 9i automatically sorted result rows by the grouping expression

Lab 6.2

```
SELECT type, AVG(length)
FROM article
GROUP BY type;
```

Query Result x

All Rows Fetched: 9 in 0

	TYPE	AVG(LENGTH)
1	LAW	1955
2	EXP	2031.57142857142...
3	ITV	1994
4	ADV	2190.5
5	FMK	2395
6	POL	2225.5
7	BUS	1398.66666666666...
8	INT	1793.5
9	TEC	2222

6

GROUP BY Clause

- Any column in the SELECT clause that is not an aggregate function must be in the GROUP BY clause
 - error ORA-00979: not a GROUP BY expression

```
SELECT type, length, AVG(length)
FROM article
GROUP BY type;
```

Query Result x

SQL | Executing:SELECT type, le

ORA-00979: not a GROUP BY expression
00979. 00000 - "not a GROUP BY expression"
*Cause:
*Action:
Error at Line: 1 Column: 13

GROUP BY Practice

- Show each section and the number of students enrolled in the section

	SECTION_ID	ENROLLMENT
1	80	1
2	81	3
3	82	2
4	83	2
5	84	2
6	85	5
7	86	6
8	87	7
9	88	5
10	89	12
11	90	4

Multi-Level GROUP BY Practice

- How many sections of each course are scheduled in each classroom? Sort primarily by course and secondarily by classroom.

	COURSE_NO	LOCATION	# Sections
1	10	L214	1
2	20	L210	2
3	20	L214	1
4	20	L509	1
5	25	L210	2
6	25	L214	1
7	25	L509	2

HAVING Practice

- Show each section and the number of students enrolled in the section. Show only sections with fewer than 7 students enrolled. Sort the rows by enrollment with highest enrollment on top.

SECTION_ID	ENROLLMENT
1	141
2	86
3	155
4	148
5	85
6	88
7	126
8	95
9	105
10	92

Using WHERE and HAVING Together

- WHERE specifies which rows are initially selected from the table to participate in a query
 - executed before GROUP BY

- HAVING specifies which groups to display results for
 - executed after GROUP BY

- For articles published after the 1980's, show how many articles there are and the average length, but only for types whose average length is at least 2000 words.

```
SELECT type, COUNT(type), ROUND(AVG(length),3)
FROM article
WHERE issue >= TO_DATE('01/01/1990', 'mm/dd/yyyy')
GROUP BY type
HAVING AVG(length) > 2000
ORDER BY type;
```

Query Result: x

All Rows Fetched: 5 in 0.016 seconds

TYPE	COUNT(TYPE)	ROUND(AVG(LENGTH),3)
1 ADV	1	3285
2 EXP	3	2122.333
3 LAW	2	2303.5
4 POL	2	3981

Clause Order

Select
From
Where
Group By
Having
Order By
