

SELECT - Aggregates	
SQL standard defines five aggregate functions:	
 SUM(expr) sum of values in specified column AVG(expr) average of values in specified column MIN(expr) smallest value in specified column MAX(expr) largest value in specified column 	
 COUNT(expr) number of values in specified columnumber of rows of query results Note that an expression can be used in place of a 	mn
column name inside an aggregate function	
Keith Tang COMP2714	2

SELECT - Aggregates Operates on a single table column, returns single value COUNT, MIN, and MAX apply to both numeric and non-numeric fields (character sets and collation order) SUM and AVG used on numeric fields only Use DISTINCT before column name to eliminate duplicates No effect with MIN/MAX May have effect on SUM/AVG Each function eliminates NULLs first and operates only on remaining non-null values, except COUNT(*) COUNT(*) counts all rows of a table, regardless of NULLs or duplicate values

Aggregates - NULLs Beware NULLs can cause subtle problems for SUM() and AVG() ◆ SELECT (SUM(price) - SUM(cost)), SUM(price - cost) FROM titles; price cost 15.00 10.00 25.00 20.00 120.00 NULL 80.00 70.00 COMP2714 Keith Tang

SELECT - Aggregates Aggregate functions can be used ONLY in SELECT and HAVING clauses If SELECT includes an aggregate function and there is no GROUP BY clause, SELECT cannot reference a column outside an aggregate function This is illegal: SELECT type, SELECT SUM(sales) FROM titles: FROM titles: Kelth Tang COMP2714 5

SELECT - Aggregates Aggregate functions can be used ONLY in SELECT and HAVING clauses ◆ If SELECT includes aggregate function(s) and references column(s) outside an aggregate function -> These columns must be in the GROUP BY This is illegal: This is OK: SELECT type, SELECT type, SUM(sales) SUM(sales) FROM titles; FROM titles GROUP BY type; Keith Tang COMP2714

SELECT - Aggregates Aggregate functions can be used ONLY in SELECT and HAVING clauses Trying to find those titles with sales above the average sale ◆This is illegal: This is OK: SELECT title_name SELECT title_name FROM titles FROM titles WHERE sales > WHERE sales > AVG(sales); (SELECT AVG(sales) FROM titles); COMP2714 Keith Tang

COUNT(expr) and COUNT(*) SELECT COUNT(title_id) AS "COUNT(title_id)", COUNT(price) AS "COUNT(price)", COUNT(*) AS "COUNT(*)" FROM titles; COUNT (title_id) COUNT (price) COUNT(*) 13 12 13

SELECT COUNT(price) SUM(price) AVG(price) FROM titles;	e) AS "COUNT(DIS s) AS "SUM(price)", AS "AVG(price)" SUM(price) AVG(price) 220.65 18.3875 COUNT(DISTINCT) SUM	COUNT(DISTIN AS "COUNT(SUM(DISTINCT AS "SUM(DI AVG(DISTINCT AS "AVG(DI FROM titles;	CDISTINCT)", [price) ISTINCT)", [price) ISTINCT)"
13		187.71	18.7710

Use of DISTINCT	
SELECT COUNT(au_id) AS "COUNT(au_id)"	COUNT(au_id)
FROM title_authors;	17
SELECT DISTINCT COUNT(au_id) AS "DISTINCT COUNT(au_id)" FROM title_authors;	DISTINCT COUNT(au_id)
SELECT COUNT(DISTINCT au_id) AS "COUNT(DISTINCT au_id)"	COUNT(DISTINCT au_id)
FROM title_authors; Keith Tang COMP271	14 10

USe of MIN, MAX Find minimum, maximum SELECT MIN(price) AS "Min price", MAX(price) AS "Max price", MAX(price) - MIN(price) AS "Range" FROM titles; Min price Max price Range 6.95 39.95 33.00

Use of AVG, SUM	
♦Find average, sum	
SELECT AVG(sales) AS "Avg.Sale" SUM(sales) AS "Total Sale FROM titles	
WHERE type = 'biography';	
Avg.Sale Total Sales	
537173.667 1611521	
Keith Tang COMP2714	12

GROUP BY

- ◆Use GROUP BY clause to get sub-totals
- ◆ If WHERE clause is used with GROUP BY
 - WHERE is applied first
 - Then groups are formed from rows satisfying WHERE conditions
- - Two nulls considered equal for purpose of **GROUP BY**
- ♦ How many GROUPs with no GROUP BY?

COMP2714

300000000000000000000000000000000000000		
GROUP BY		
SELECT au_id, COUNT(*) AS "num	_books"	
FROM title_authors GROUP BY au_id;	au_id r A01	num_books
List the number of books each author wrote or co-wrote	A02 A03 A04 A05 A06	4 2 4 1 3
Keith Tang COMP2	714	14

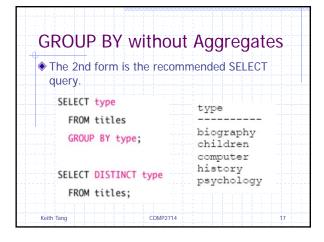
GROUP BY

- ◆ SELECT and GROUP BY closely related:
 - All column names in SELECT list must appear in GROUP BY clause, unless column name is used
 - in an aggregate function
 - Each item in SELECT list must be singlevalued per group
 - What is wrong with the following? SELECT type, pub_id, COUNT(*) FROM titles GROUP BY type;

Keith Tang

--Illegal

Multiple Grouping Columns			
List the number of bo	ooks of ea	ch type for eac	h publishe
	pub_id	type	COUNT(*)
ELECT	P01	biography	3
<pre>pub_id, type,</pre>	P01 P02 P03	history computer history	1
COUNT(*) AS "COUNT(*)"		biography psychology	1
FROM titles GROUP BY pub_id, type	P04	children	
ORDER BY pub_id ASC, "COU	NT(*)" DES	c;	
Keith Tang	COMP2714		16



Restrict Groupings – HAVING HAVING clause is designed for use with GROUP BY to restrict groups that appear in final result table Not to be confused with WHERE WHERE filters individual rows from FROM HAVING filters grouped rows from GROUP BY Use WHERE wherever possible; not HAVING Column names in HAVING clause Can reference any items in SELECT

Use of HAVING		
◆List the number of bo	ooks written	(or co-
written) by each auth	nor who has	written
three or more books		
SELECT	au_id num	_books
au_id,		
COUNT(*) AS "num_books"	A01 A02	3
FROM title_authors	A04	4
GROUP BY au_id	A06	3
HAVING COUNT(*) >= 3;		
HAVING COUNT(+) >= 5;		

