Basic Select (chap7.4)

Select columns	SELECT *colname1*, *colname2* FROM **Table1**	
Condit-ional select	SELECT *colname1*, *colname2* FROM **Table1** WHERE *colname1* = *value1*	Other operators: >, <, >=, <=, <>(or !=)
Select based on logical conditions	SELECT *colname1*, *colname2* FROM **Table1** WHERE *colname1* = *value1* AND *colname2* > *value2*	Other logical operators: OR, NOT
Select based on value in range	SELECT *colname1*, *colname2* FROM **Table1** WHERE *colname1* BETWEEN *lower limit* AND *upper limite*	Smaller value must be first in Oracle

Basic Select (chap7.4) (cont)

Select null values	SELECT *colname1*, *colname2* FROM **Table1** WHERE *colname3* IS NULL	
Select based on fuzzy string matching	SELECT *colname1*, *colname2* FROM **Table1** WHERE *colname1* LIKE *'%word%'*	'%' matches one or more characters, '_' matches exactly one character; LIKE is case sensitive in Oracle
Select based on value in a list	SELECT *colname1*, *colname2* FROM **Table1** WHERE *colname1* in (*value1, value2*)	all members in list must be same type; can be used in conjuction with subqueries

Intermediate select

Order results of select query	SELECT *colname1*, *colname2* FROM **Table1** ORDER BY *colname1*, *colname2*	

Intermediate select (cont)

List unique values only	SELECT DISTINCT *colname1*, *colname2* FROM **Table1**	In Oracle, Null value will be included as one distince value; use ORDER BY *colname* NULLS FIRST to place nulls at top of list

Aggregate functions

Count NON-NULL values of an attribute	SELECT COUNT(*colname1*) FROM **Table1**	
Count unique NON-Null values	SELECT COUNT(DIS-TINCT(*colname1*)) FROM **Table1**	
Select maximum and minimum values in an attribute	SELECT MAX/MIN(*colname1)* FROM **Table1**	Yields ONLY ONE VALUE; can only be used inside a SELECT query

Aggregate functions (cont)

Select sum and average of an attribute	SELECT SUM/AVG(*colname1)* FROM **Table1**	
Use aggregate functions on computed values	SELECT SUM/AVG/M IN/MAX(*colname1* - c FROM **Table1**	

*Group by clause

Calculate aggregate function based on group membership	SELECT *colname1*, MAX(*colname2*) FROM **Table1** GROUP BY *colname1*	G B b c w A fu G B m ir c ir c o th th a fu c ir c th n s a

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

*Group by clause (cont)

Filter the result of GROUP BY clause with HAVING clause	SELECT *colname1*, MAX(*colname2*) FROM **Table1** GROUP BY *colname1* HAVING MAX(*colname2*) > *value1*	In Oracle, column alias can be used for the result of the aggregate function column

Relational Set Operators

Get all records from two relations excluding duplicates	Set1 UNION Set2
Get all records from two relations including duplicates	Set1 UNION ALL Set2
Get all records in both relations	Set1 INTERSECT Set2
Get all recods in relations 1 but not in relation 2	Set1 MINUS Set2

Utility Functions

Formatting datetime into character	TO_CHAR(date-_value, fmt)	MONTH; MON; MM; D(DOW);DD-(DOM); DAY;YYYY;YY

Utility Functions (cont)

Formatting character into datetime	TO_DATE(char-acter_s-tring)	Same as above
Current date	SYSDATE	
Adding months to a date	ADD_MONTH-S(date_value, n)	
Absolute value	ABS()	
Round falue to a specific precision	ROUND(numeri-c_value, p)	
Ceiling value	CEIL(numeric-_value)	
Floor value	FLOOR(numeri-c_value)	
Concat-enating String	*string1* || *string2*|| *string3*	
Uppercase	UPPER(string-_value)	
Lowercase	LOWER(string-_value)	

Utility Functions (cont)

Subset a string	SUBSTR(string_value, starting_index, ending-_index)
length of a string	LENGTH(string_value)

Join

Inner joins	INNER JOIN
Join two table on common attributes but retain rows in the first table without a match	LEFT JOIN
Join two table on common attributes but retain rows in the second table without a match	RIGHT JOIN
Join two table on common attributes but retain unmatched rows	FULL JOIN

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com