Cheat Sheet Exam 2

**Where Clause**

SELECT fname, lname, salary FROM patient WHERE salary <>30000; Select people who salary is **not equal** to 30,000 use **!= or <>**

SELECT fname, lname, salary , city FROM patient WHERE city='Davis'; Selects peoples whos city is Davis case sensitive

SELECT fname, lname, salary , city FROM patient WHERE city> 'Davis' ; Selects city that start with a letter after Davis

**Dates**

SELECT fname, lname,DOB, salary FROM patient WHERE DOB>'11-FEB-1978'; Selects where date is bigger than date mentioned

SELECT fname, lname,DOB, salary FROM patient WHERE DOB>TO\_DATE('02/11/1978','mm/dd/yyyy'); not in correct format use toDate

**ToUpper and ToLower**

SELECT fname, lname,city FROM patient WHERE UPPER(city)='DAVIS'; Used to bypass case sensitivity.

**And Between**

SELECT fname, lname,DOB, salary FROM patient WHERE salary>30000 and salary <80000; used and to check between 2 values

SELECT fname, lname, DOB, salary FROM patient WHERE salary BETWEEN 30000 AND 80000; used to check between 2 values

**Or In**

SELECT fname, lname, city FROM patient WHERE city='Davis' OR city='Sacramento' OR city='Chico'; if the city is in one of these cities

SELECT fname, lname, city FROM patient WHERE city IN ('Davis','Sacramento','Chico'); Uses in to see if city is in one of those cities

Can do the same for numbers example salary can also use not to make sure its not in there

WHERE clause can contain multiple types of operators, you need to understand the order in which they’re resolved: • Arithmetic operations are

solved first. Comparison operators are solved next. Logical operators have a lower precedence and are evaluated last in the order NOT, AND, and OR.

**Any or All**

SELECT fname, lname, salary FROM patient WHERE salary >ANY(30000,40000); where salary is greater than any of the values in the parenthesis

SELECT fname, lname, salary FROM patient WHERE salary >ALL(30000,40000); where salary is greater than all the values in the parenthesis

**Like Clause**

SELECT fname, lname,DOB, city FROM patient WHERE city LIKE 'D%'; select all cities where the city starts with D

SELECT fname, lname,DOB, city FROM patient WHERE city LIKE '%D%'; select all cities that have the letter D

SELECT fname, lname,DOB, city FROM patient WHERE UPPER(city) LIKE '%S'; cities that end with the letter S

SELECT fname, lname,DOB, city FROM patient WHERE city LIKE 'D\_vis'; the underscore can be any letter Davis Devis …etc

**Null**

SELECT fname, lname, city FROM patient WHERE city IS NULL; Correct Null Notation

**Creating tables with select statements**

CREATE TABLE patient\_temp AS SELECT fname, lname, salary, city FROM patient WHERE city='Mavis'; making a table using the results of another table.

CREATE TABLE patient\_temp AS SELECT fname, lname, salary\*2 new\_salary FROM patient WHERE salary >30000; Needs alias if doing calc

**Updating tables using Update**

UPDATE patient SET fname=’Bill’, lname=’Bob’, salary=10000 WHERE patient\_id=111;

**Sorting**

SELECT patient\_id,fname,lname,city CITYNAME FROM patient ORDER BY city; orders by Ascending by default so by a-z order or order by CityName

SELECT patient\_id,fname,lname,city CITYNAME FROM patient ORDER BY city DESC; orders by z-a order

When sorting in ascending order, values are listed in this order: Blank and special characters Numeric values Character values ( uppercase first) NULL values

SELECT patient\_id,fname,lname,city CITYNAME FROM patient ORDER BY fname DESC, lname; Orders by fname in descending order first then in lname ascending order by default

SELECT patient\_id,fname,lname,city CITYNAME FROM patient ORDER BY 4; will order by city since it’s the 4th value or by cityname since it’s the alias

**NVL**

SELECT NVL(salary,0) pay FROM patient ORDER BY pay; NVL replaces null with 0 and then it is ordered by pay

SELECT city, SUM(salary) FROM patient GROUP BY city; adds all the salaries and groups by city. Can group by state

SELECT state, SUM(salary) FROM patient GROUP BY state; --- SELECT city, gender, SUM(salary) FROM patient GROUP BY city, gender; Groups by city first then gender then display sum salary

**SUM** SELECT SUM (salary) FROM patient; normal syntax --- SELECT SUM (DISTINCT salary) FROM patient; a number that is a results of all distinct salaries being added

**AVG** SELECT AVG (salary) FROM patient; basic syntax-- SELECT AVG (DISTINCT salary) FROM patient;--- SELECT city, AVG (salary) FROM patient GROUP BY city ORDER BY 1; or order by city

SELECT city, AVG (nvl(salary,0)) FROM patient WHERE UPPER(city) <> 'RENO'GROUP BY city HAVING AVG(salary)>20000 ORDER BY 1; filters data on where clause then it takes left over records and does a grouping on average salary greater then 20000. If a city does not have a salary it will be replaced with 0

**Count** SELECT COUNT (\*) FROM patient;--counts number of rows. SELECT COUNT (city) FROM patient; counts rows if not null -- SELECT city, COUNT (city) FROM patient GROUP BY city HAVING COUNT(\*) >1; counts the cities based on categories doesn’t exclude nulls and the count has to be greater than 1.

**Max** SELECT MAX (salary) FROM patient; --displays the highest salary. SELECT city, MAX (salary) FROM patient; --displays the highest slaaray in the city

**MIN** SELECT MIN (salary) FROM patient; --lowest salary is displayed -- SELECT city, MIN (salary), COUNT(\*) FROM patient GROUP BY city; Display the lowest salary for each city category and display the number of records in each group.

**DATE** SELECT min(DOB), max (DOB), count(DOB), count (DISTINCT DOB) FROM patient; --selects the youngest, oldest and the number or records no nulls and not dupes. **Find Age** TRUNC(Months\_between(Sysdate,dob)/12)

**Subquireis**

SELECT fname, lname FROM patient WHERE patient\_id IN(

SELECT patient\_id FROM patient\_disease WHERE disease\_id=(

SELECT disease\_id FROM disease WHERE disease\_desc='Cancer'))

SELECT disease\_desc FROM disease WHERE disease\_id IN (

SELECT disease\_id FROM patient\_disease WHERE patient\_id=(

SELECT patient\_id FROM patient WHERE fname=’john’ and lname=’Doe' ));

Multiple Column

SELECT patient\_id FROM patient WHERE (fname,lname) IN (

SELECT fname,lname FROM special\_names);

**Create Table using subquries**

CREATE TABLE NEW\_TABLE1 AS SELECT patient\_id FROM patient WHERE

(fname,lname) IN (SELECT fname,lname FROM table\_names);

**Creating new Table from old table** CREATE TABLE new\_table AS SELECT \* FROM old\_table; --copies everyhitng --- CREATE TABLE suppliers AS SELECT \*FROM companies WHERE (company\_id <5000);

**UPDAte and Delete**

UPDATE patient SET salary=salary\*2 WHERE patient\_id IN (

SELECT patient\_id FROM patient\_disease WHERE disease\_id=(

SELECT disease\_id FROM disease WHERE disease\_desc='Cancer')) ;

DELETE FROM patient\_disease WHERE disease\_id IN (

SELECT disease\_id FROM disease WHERE disease\_desc='Cancer');