SQL RETRIEVAL EXAMPLES:

(oRGANIZE PREDICATES IN THE PROPER ORDER)

(WHERE COL1 < 10 AND COL2 = 1000 - HIGH TO LOW). Use Select Count(\*)

from staff where dept = 10 for example to find the total number of values for dept. Do this for each criteria to identify the HIGH to LOW order. Assuming no indexes available.

1. Give names of all managers making less than $20,000 and are in department

10.

1. For all employees in department 66 making a commission less than 350.00,

and their “total annual earnings” are > 15000.00 give their name, job function

and total annual earnings (Salary and Commissions).

3. Give the name and department number of each employee not in department 15 who make more than 20000.00 salary.

1. List all information about employee IDs up to and including 100 who would make more than 20000.00 if they are given a 15% salary increase or would

make more than 350.00 commission if a 100.00 bonus were added to their

current commission.

BUILT-IN FUNCTIONS ExERCISES

## EXERCISE 2

1) What is the total number of departments in the staff table.? See ADVEX21.sql

RESULTS:



(1 ROW SELECTED)

2) Print out the maximum salary and minimum salary for department 10.

RESULTS: ADVEX22



(1 ROW SELECTED)

3) List the average salary for CLERKS. See ADVEX23.sql

RESULTS:



(1 ROWS SELECTED)

4) What is the maximum salary for employees with less than 5 years of service? See ADVEX24.sql

RESULTS:



(1 ROWS SELECTED)

5) What is the total years of service for all employees in departments 15, 20, and 42? See ADVEX25.sql

RESULTS



(1 ROW SELECTED)

THE HAVING CLAUSE EXERCISES:

**EXERCISE 3**

1) For each department whose average salary is greater than $18,000, List the   
 department number, average years of service, and average salary. See

ADVEX31

RESULTS:



(1 ROW SELECTED)

2) List each department number where all employees have at least 5 years of   
 service. See ADVEX32.sql

RESULTS:



(4 ROWS SELECTED)

3) List each department number having a total commission of at least $1200

See ADVEX33.sql

RESULTS:



(6 ROWS SELECTED)

# LAB 3 - MORE ABOUT SELECTS



Using SQL\*PLUS or SQL Developer or Toad, create and execute the following queries using the staff, org and applicant tables. See Lab3.txt for answers

1. How many clerks does the company currently have?

2. List the name, salary and commission of each employee sorted by

department

1. List the job functions and total salaries by job of all employees working in departments 20, 42, 51, and 84.
2. Give the average salary, total salary, average commission and total commission for each department but only if they have a total commission greater than 1200.
3. List all sales personnel by descending total earnings (Salary Commission) who belong to departments 15, 20, 42 & 84 who also have a salary greater than 15000.

6. List the names of all employees whose name contains the characters 'ON'.

7. Display all employees whose weekly salary is greater than $300.00.

8. Display all employees whose commission is at least 15% of total earnings.

1. List id, name, salary, and commission for all employees except departments   
    38 and 41. Order by Department

10. List every department with at least one employee with a null commission.   
 List a department number only one time.

1. List the name, salary and department name of each employee. Use either a

case or a decode statement to produce the department name.

LAB 3 ANSWERS



Using SQL\*PLUS, SQL Developer or Toad, create and execute the following queries using the staff, org and applicant tables. See Lab3.txt for answers

1. How many clerks does the company currently have?

SQL> select count(\*) “# of CLERKS” from staff where job = ‘CLERK’

2. List the name, salary and commission of each employee sorted by

Department

SQL> select dept, name, salary, comm

From staff order by dept;

1. List the job functions and total salaries by job of all employees working in departments 20, 42, 51, and 84.

SQL> select job, sum(salary) from staff

Where dept in (20, 42, 51, 84)

Group by job

1. Give the average salary, total salary, average commission and total commission for each department but only if they have a total commission greater than 1200.

SQL> select dept, avg(salary), sum(salary), avg(comm), sum(comm.)

From staff

Group by dept

Having avg(comm) > 1200

1. List all sales personnel by descending total earnings (Salary Commission) who belong to departments 15, 20, 42 & 84 who also have a salary greater than 15000.

SQL> SELECT DEPT, NAME, JOB, SALARY + COMM

From staff

Where dept in (15, 20, 42, 84) and

Job = ‘SALES’ and

Salary > 15000.00

6. List the names of all employees whose name contains the characters 'ON'.

SQL> select name from staff where name like ‘%ON%’;

7. Display all employees whose weekly salary is greater than $300.00.

SQL> select dept, name, salary/52 “Weekly Salary”

From staff

Where salary/52 > 300.00 – or (salary > 300.00 \* 52)

1. Display all employees whose commission is at least 15% of total earnings.

SQL> select dept, name, salary, comm,

(salary + comm) \* .15 “15% of total earnings”

From staff

Where comm > (salary + comm) \* .15

1. List id, name, salary, and commission for all employees except departments 38 and 41. Order by Department.

SQL> select dept, id, name, salary, comm

From staff

Where dept not in (38, 41)

Order by dept

10. List every department with at least one employee with a null commission.   
 List a department number only one time.

SQL> select distinct dept from staff

Where comm Is null;

11. List the name, salary and department name of each employee. Use either a

case or a decode statement to produce the department name.

SQL> select name, salary, decode(dept,10,’CORPORATE’,

15,’INVENTORY’,

20,’RESEARCH’,

38, ‘PRODUCTION’,

42, ‘ACCOUNTING’,

51, ‘WAREHOUSE’,

66, ‘SOUTHWEST’,

81, ‘NORTHWEST’)

FROM STAFF

# LAB 4

# Using SQL\*PLUS, SQL Developer or Toad

1. Execute the script Bigpart.sql as the user hr. Ensure you have the tablespace User\_Data, Tools and Jerindex created (must be a DBA) before executing the script. Ask the instructor for assistance as required. Write a select statement against the big table or the table the instructor identifies for you.

select \* from big

where bigno < 5000;

1. As the hr user, Write a query against the V$SQLAREA that shows the

number of executions,

the number of buffer gets, the number of parse calls, and the first 20

characters of TEXT for all statements that execute more than 3000 times.

3. Login to a separate SQL\*PLUS or SQL DEVELOPER Session as hr and Determine approximately how many rows are being stored in each block of

the EMP table & BIG table by writing an adhoc query using ROWID.

See tunex8.sql

4. Insert a duplicate row into the org table.

insert into org values (10,’HEAD OFFICE’,160,’CORPORATE’,’NEW YORK’);

5. DELETE the row you just inserted using the ROWID.

6. Retrieve Lu from the staff table using ROWID.

7. Determine the approximate number of blocks in the BIG table.

(HINT: look at tunex8.sql)

8. Create statistics on the user SCOTT’s tables so that they will run CBO.8. Provide hints to the optimizer for executing SQL statements.

1. Write a query to retrieve the id, employee name, years

of service, and salary from the staff table for all employees

in department 15. Use COST based hints. (SEE lab4\_8.sql)

SELECT /\*+ FIRST\_ROWS \*/ \* FROM STAFF WHERE DEPT = 15;

2. Write another query on the ORD table which provides a hint for using

an index called ord\_index .

SELECT /\*+ INDEX (ORD.ORD\_INDEX) \*/ \* FROM ORD

WHERE ORDID > 200;

9. Using SQL Developer go to the Reports Section | Data Dictionary | All

Objects and identify all invalid objects for the user HR and the SCOTT user.

10. Login to SQL\*PLUS as the hr user (another new session). Update the staff table so that everyone has a salary of 77,000.00. In another SQL\*PLUS session as the hr user, update the id of employee id 340 with a comm of 500. What happens when you attempt to execute this query? Then In the Reports Section of SQL Developer | Database Administration identify all the cursors open for the user hr.

11 In SQL Developer | Database Administration, Locks identify all the user and sessions who currently have locks. (Optionally) identify all users who are blocking other users.

1. Using SQL Developer | Reports | Database Administration | Sessions | Sessions kill anyone who is blocking another user.