**Practice 1: Simple select statements**

1. Initiate a SQL\*Plus session using the user ID and password provided by the instructor.

2. SQL\*Plus commands access the database.  
 True/False

Answer:True

3. Will the SELECT statement execute successfully?  
 True/False

**SQL> SELECT ename, job, sal Salary  
 2 FROM emp;**

**Answer:True**

4. Will the SELECT statement execute successfully?  
 True/False

**SQL> SELECT \*  
 2 FROM salgrade;**

**Answer True**

5. There are four coding errors in this statement. Can you identify them?

**SQL> SELECT empno, ename  
 2 salary x 12 ANNUAL SALARY  
 3 FROM emp;**

**First Error: A star is missing**

**Second Error:Quotes missing around annual salary**

**Third: The comma missing after ename**

6. Show the structure of the DEPT table. Select all data from the DEPT table.

Name Null? Type

----------- -------- ------------

DEPTNO NOT NULL NUMBER(2)  
 DNAME VARCHAR2(14)  
 LOC VARCHAR2(13)

DEPTNO DNAME LOC  
 ------ ---------- -------------  
 10 ACCOUNTING NEW YORK  
 20 RESEARCH DALLAS  
 30 SALES CHICAGO  
 40 OPERATIONS BOSTON

DESC DEPT

Select \* FROM DEPT;

7. Show the structure of the EMP table. Create a query to display the name, job, hire date,  
 and employee number for each employee, with employee number appearing first. Save  
 your SQL statement to a file named p1q7.sql.

select employee\_id,first\_name||last\_name as name, job\_id, hire\_date

FROM employees;

**Practice 2 :Creating and managing tables**

1. Create the DEPARTMENT table based on the following table instance chart. Enter thesyntax in a script called p2q1.sql, then execute the script to create the table. Confirm that the table is created.

Name Null? Type   
 ---------- -------- -----------

ID NUMBER(7)  
 NAME VARCHAR2(25)

CREATE TABLE Department

(

ID Number(7) ,Name VARCHAR2(25)

);

2. Populate the DEPARTMENT table with data from the DEPT table. Include only columns that you need.

Select \* from Department;

3. Create the EMPLOYEE table based on the following table instance chart. Enter the syntax in a script called p2q3.sql, and then execute the script to create the table. Confirm that the table is created.

Name Null? Type

------------- -------- ------------

ID NUMBER(7)  
 LAST\_NAME VARCHAR2(25)  
 FIRST\_NAME VARCHAR2(25)  
 DEPT\_ID NUMBER(7)

CREATE TABLE EMPLOYEE

(

ID Number(7) ,LAST\_NAME VARCHAR2(25),FIRST\_NAME VARCHAR2(25),DEPT\_ID NUMBER(7)

);

4. Modify the EMPLOYEE table to allow for longer employee last names. Confirm your  
 modification.

Name Null? Type

------------- -------- -----------

ID NUMBER(7)  
 LAST\_NAME VARCHAR2(50)  
 FIRST\_NAME VARCHAR2(25)  
 DEPT\_ID NUMBER(7)

Alter Table Employee

Modify Last\_Name VARCHAR2(50)

5. Confirm that both the DEPARTMENT and EMPLOYEE tables are stored in the data  
 dictionary. (*Hint:* USER\_TABLES)

TABLE\_NAME  
 -------------------------  
 DEPARTMENT  
 EMPLOYEE

6. Create the EMPLOYEE2 table based on the structure of the EMP table. Include only theEMPNO, ENAME, and DEPTNO columns. Name the columns in your new table ID,LAST\_NAME, and DEPT\_ID, respectively.

CREATE TABLE EMPLOYEE2

(

ID Number(7) ,LAST\_NAME VARCHAR2(25),DEPT\_ID NUMBER(7)

);

7. Drop the EMPLOYEE table;

Drop table Employee

8. Rename the EMPLOYEE2 table to EMPLOYEE.

Select \* from Employee2 AS Employee;

9. Add a comment to the DEPARTMENT and EMPLOYEE table definitions describing the tables. Confirm your additions in the data dictionary.

COMMENT ON TABLE Department IS ‘Confirm your additions in the data dictionary’

COMMENT ON TABLE Employee IS ‘Confirm your additions in the data dictionary’

10. Drop the LAST\_NAME column from the EMPLOYEE table. Confirm your modification by checking the description of the table.

Alter table Employee

Drop column Last\_Name;

11. Create the EMPLOYEE2 table based on the structure of the EMP table. Include only the EMPNO, ENAME, and DEPTNO columns. Name the columns in your new table ID,  
 LAST\_NAME, and DEPT\_ID, respectively. Mark the DEPT\_ID column in the EMPLOYEE2 table as UNUSED. Confirm your modification by checking the description of the table.

CREATE TABLE EMPLOYEE2

(

LAST\_NAME VARCHAR2(25),DEPT\_ID NUMBER(7)

);

12. Drop all the UNUSED columns from the EMPLOYEE2 table. Confirm your modification by checking the description of the table.

Alter table Employee2

Drop columns UNUSED;

**Practice 3:Constraints**

1. Add a table-level PRIMARY KEY constraint to the EMPLOYEE table using the ID column. The constraint should be named at creation.

Alter Table Employee

Add primary key(P\_ID)

Hint: The constraint is enabled as soon as the ALTER TABLE command executes successfully.

2. Create a PRIMARY KEY constraint on the DEPARTMENT table using the ID column. The constraint should be named at creation.

Hint: The constraint is enabled as soon as the ALTER TABLE command executes successfully.

Alter Table Department

Add primary key (P\_ID)

3. Add a foreign key reference on the EMPLOYEE table that will ensure that the employee is not assigned to a nonexistent department.

ALTER TABLE Employee

ADD FOREIGN KEY (P\_Id)

REFERENCES department(P\_Id)

4. Confirm that the constraints were added by querying USER\_CONSTRAINTS. Note the types and names of the constraints. Save your statement text in a file called p11q4.sql.

CONSTRAINT\_NAME C  
 ----------------------- --  
 DEPARTMENT\_ID\_PK P  
 EMPLOYEE\_ID\_PK P  
 EMPLOYEE\_DEPT\_ID\_FK R

5. Display the object names and types from the USER\_OBJECTS data dictionary view for EMPLOYEE and DEPARTMENT tables. You may want to format the columns for readability. Notice that the new tables and a new index were created.

OBJECT\_NAME OBJECT\_TYPE  
 ---------------- -----------  
 DEPARTMENT TABLE  
 DEPARTMENT\_ID\_PK INDEX  
 EMPLOYEE TABLE  
 EMPLOYEE\_ID\_PK INDEX

**Practice 4: DML**

Insert data into the MY\_EMPLOYEE table.

1. Run the lab9\_1.sql script to build the MY\_EMPLOYEE table that will be used for the lab.

Insert Into My\_Employee

Values('1','Henry','Tom','tom1',50000');

2. Describe the structure of the MY\_EMPLOYEE table to identify the column names.  
 Name Null? Type  
 ------------ --------- ------  
 ID NOT NULL NUMBER(4)  
 LAST\_NAME VARCHAR2(25)  
 FIRST\_NAME VARCHAR2(25)  
 USERID VARCHAR2(8)  
 SALARY NUMBER(9,2)

DESC MY\_EMPLOYEE Column names are ID,LAST\_NAME,FIRST\_NAME, USERID, SALARY

3. Add the first row of data to the MY\_EMPLOYEE table from the following sample data. Do not list the columns in the INSERT clause.

Insert Into My\_Employee

Values('1','Henry','Tom','tom1',50000');

4. Populate the MY\_EMPLOYEE table with the second row of sample data from the preceding list. This time, list the columns explicitly in the INSERT clause.

INSERT INTO My\_Employee (ID,Last\_Name,First\_Name,USERID,SALARY)

VALUES (3,Smith,Dara,Dara321,70000);

5. Confirm your addition to the table.

ID LAST\_NAME FIRST\_NAME USERID SALARY  
 --- ----------- ---------- ------ ------  
 1 Patel Ralph rpatel 795  
 2 Dancs Betty bdancs 860

Update and delete data in the MY\_EMPLOYEE table.

6. Change the last name of employee 3 to Drexler.

UPDATE MY\_EMPLOYEE

SET LAST\_NAME='Drexter'

WHERE EmployeeID=3;

7. Change the salary to 1000 for all employees with a salary less than 900.

UPDATE MY\_EMPLOYEE

SET SALARY=1000

WHERE SALARY>900;

8. Verify your changes to the table.

9.. Delete Betty Dancs from the MY\_EMPLOYEE table.

DELETE FROM MY\_EMPLOYEE

WHERE FIRST\_NAME='Betty' AND LAST\_NAME='DANCS';

10. Confirm your changes to the table.

**Practice 5: Restricting and sorting data**

1. Create a query to display the name and salary of employees earning more than $2850. Save your SQL statement to a file named p5q1.sql. Run your query.

Select name, sal From emp where sal>$2850;

2. Create a query to display the employee name and department number for employee number 7566.

Select ename, deptno From emp where empno = 7566;

3. Modify p5q1.sql to display the name and salary for all employees whose salary is not in the range of $1500 and $2850. Resave your SQL statement to a file named p2q3.sql. Rerun your query.

Select name,sal From emp where sal not in ($1500 and $2850);

4. Display the employee name, job, and start date of employees hired between February 20, 1981, and May 1, 1981. Order the query in ascending order by start date.

Select ename,job,hiredate From emp order by hiredate asc;

5. Display the employee name and department number of all employees in departments 10 and 30 in alphabetical order by name.

Select ename,deptno From emp where deptno 10 and 30, ename asc;

6. Modify p5q3.sql to list the name and salary of employees who earn more than $1500 and are in department 10 or 30. Label the columns Employee and Monthly Salary, respectively. Resave your SQL statement to a file named p2q6.sql. Rerun your query.

Select name AS Employee, salary of employee AS Monthly Salary,depno From emp where Monthly Salary>$1500 AND deptno 10 or 30;

7. Display the name and hire date of every employee who was hired in 1982.

Select name, hiredate From emp where hiredate=1982;

8. Display the name and job title of all employees who do not have a manager.

Select name, job From emp where job not in (‘manager’);

9. Display the name, salary, and commission for all employees who earn commissions. Sort data in descending order of salary and commissions.

Select name,sal,comm From emp order by commission and salary desc;

10. Display the names of all employees where the third letter of their name is an A.

Select name From emp where name=’ --A%’;

11. Display the name of all employees who have two Ls in their name and are in department 30 or their manager is 7782.

Select name From emp where name=’LL% department=30 or manager=7782;

12. Display the name, job, and salary for all employees whose job is Clerk or Analyst andtheir salary is not equal to $1000, $3000, or $5000.

Select name,job,sal From emp where job=’clerk or ‘analyst’ sal not in ($1000,$3000,or $5000);

13. Modify p2q6.sql to display the name, salary, and commission for all employees whosecommission amount is greater than their salary increased by 10%.

Select name,sal,com From emp where comm>sal+10%;

**Practice 6: Single row functions**

1. Write a query to display the current date. Label the column Date.

Date  
 ---------  
 28-OCT-97

Select Sysdate AS Date

From Dual;

2. Display the employee number, name, salary, and salary increase by 15% expressed as a whole number. Label the column New Salary. Save your SQL statement to a file named  
 p3q2.sql.

Select employee number, name,salary,

ROUND(salary \* 1.15, 0) "New Salary"

From Employee;

3. Modify your query p3q2.sql to add a column that will subtract the old salary from  
 the new salary. Label the column Increase. Rerun your query.

SELECT employee number, name, salary,

ROUND(salary \* 1.15, 0) "New Salary",

ROUND(salary \* 1.15, 0) - salary "Increase"

FROM employees;

4. Display the employee’s name, hire date, and salary review date, which is the first Monday after six months of service. Label the column REVIEW. Format the dates to appear in the format similar to “Sunday, the Seventh of September, 1981.”

SELECT employee name, hire date, ADD\_MONTHS(hiredate, 6) AS REVIEW

FROM employees;

5. For each employee display the employee name and calculate the number of months between today and the date the employee was hired. Label the column MONTHS\_WORKED. Order your results by the number of months employed. Round the number of months up to the closest whole number.

SELECT employee name

ROUND(MONTHS\_BETWEEN

(SYSDATE, hire\_date)) MONTHS\_WORKED

FROM employees

ORDER BY MONTHS\_BETWEEN(SYSDATE, hire\_date);

**Practice 7:Displaying data from multiple tables**

1. Write a query to display the name, department number, and department name for all employees.

Select employees.name department.department number.department name From employees,department;

2. Create a unique listing of all jobs that are in department 30. Include the location of department 30 in the output.

Select distinct jobs from employees where department=30;

3. Write a query to display the employee name, department name, and location of all  
employees who earn a commission.

Select employees.employee name,department.department name department.location from department,employees Order by commission;

4. Display the employee name and department name for all employees who have an *A* in their name

Select employees.employee name, department.department name from employees,department where employee name='%A%';

5. Write a query to display the name, job, department number, and department name for all employees who work in DALLAS.

Select employees.name, employees.job, department.department number,department name from employees,department where location='Dallas';

6. Display the employee name and employee number along with their manager’s name andmanager number. Label the columns Employee, Emp#, Manager, and Mgr#, respectively. Save your SQL statement in a file called p4q6.sql.

7. Create a query to display the name and hire date of any employee hired after employee Blake**.**

8. Display all employees’ names and hire dates along with their manager’s name and hire date for all employees who were hired before their managers. Label the columns Employee, Emp ,Hiredate, Manager, and Mgr Hiredate, respectively**.**

**Practice 8: Aggregating and grouping functions**

Determine the validity of the following statements. Circle either True or False.

1. Group functions work across many rows to produce one result per group.  
 True/False

True

2. Group functions include nulls in calculations.  
 True/False

False

3. The WHERE clause restricts rows prior to inclusion in a group calculation.  
 True/False

True

4. Display the highest, lowest, sum, and average salary of all employees. Label the column  
Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number. Save your SQL statement in a file called p5q4.sql.

SELECT ROUND(MAX(salary),0) "Maximum",

ROUND(MIN(salary),0) "Minimum",

ROUND(SUM(salary),0) "Sum",

ROUND(AVG(salary),0) "AverageSalary"

FROM employees;

5. Modify p5q4.sql to display the minimum, maximum, sum, and average salary for each job type. Resave to a file called p5q5.sql.

SELECT job\_id, ROUND(MAX(salary),0) "Maximum",

ROUND(MIN(salary),0) "Minimum",

ROUND(SUM(salary),0) "Sum",

ROUND(AVG(salary),0) "AverageSalary"

FROM employees

Group by job\_id

6. Write a query to display the number of people with the same job.

SELECT employee\_id, COUNT(\*)

FROM employees

Group by job\_id;

**Practice 9 : Subqueries**

Write a query to display the number of people with the same job.

SELECT employee\_id, COUNT(\*)

FROM employees

Group by job\_id;

* Create a query to display the employee number and name for all employees who earn more than the average salary. Sort the results in descending order of salary
* Select employee number,employee name from employees Where SALARY>Average Salary;
* Write a query that will display the employee number and name for all employees who work in a department with ancontains a *T*y employee whose name

Select employee name,employee number from employees Where employee name='%TY%';

* Display the employee name, department number, and job title for all employees whose department location is Dallas
* Select employee name,department number, job title from employees Where department location='Dallas';
* Display the employee name and salary of all employees who report to King.

Select employee name,salary from employees where manager='King';

* Display the department number, name, and job for all employees in the Sales department.

Select department number, name,job from employees where department='Sales';

**Practice 10 : Multiple column subqueries**

* Write a query to display the name, department number, and salary of any employee whosedepartment number and salary match the department number and salary of any employee who earns a commission
* Display the name, department name, and salary of any employee whose salary and commission match the salary and commission of any employee located in Dallas.
* Create a query to display the name, hire date, and salary for all employees who have both the same salary and commission as Scott.

**Note:** Do not display SCOTT in the result set.

* Create a query to display the employees that earn a salary that is higher than the salary of  
  all of the clerks. Sort the results on salary from highest to lowest**.**