**Practice 1**

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

Ans

**2. *i*SQL\*Plus commands access the database.**

Answer: false

**3. The following SELECT statement executes successfully:**

SELECT last\_name, job\_id, salary AS Sal  
 FROM employees;

Answer: true

**4. The following SELECT statement executes successfully**:

SELECT \*

FROM job\_grades;

Answer: false

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

SELECT employee\_id, last\_name

sal x 12 ANNUAL SALARY

FROM employees;

Answer: (1: , 2: \* 3 :ANNUAL SALARY 4: sal)

**6. Show the structure of the DEPARTMENTS table. Select all data from the table**

Answer 1: Describe departments

Answer 2: select \* from departments;

**7. Show the structure of the EMPLOYEES table. Create a query to display the last name, job code, hire date, and employee number for each employee, with employee number appearing first. Provide an alias STARTDATE for the HIRE\_DATE column. Save your SQL statement to a file named lab1\_7.sql.**

Answer 1: **Describe employees**

**8. Run your query in the file lab1\_7.sql**

Answer 2: **select last\_name,job\_id,hire\_date"startdate",employee\_id from employees;**

**9. Create a query to display unique job codes from the EMPLOYEES table.**

Answer : Select distinct(job\_id) From employees;

**10. Copy the statement from lab1\_7.sql into the *i*SQL\*Plus Edit window. Name the column headings Emp #, Employee, Job, and Hire Date, respectively. Run your query again.**

Answer : select employee\_id"#emp",last\_name"employee",job\_id"job",hire\_date"hire date"

from employees;

**11. Display the last name concatenated with the job ID, separated by a comma and space, and name the column Employee and Title.**

Answer : **select last\_name||', '||job\_id "employees and title from employees;**

**12. Create a query to display all the data from the EMPLOYEES table. Separate each column by a comma. Name the column THE\_OUTPUT.**

Answer : **select employee\_id||','||first\_name||','||last\_name||','||email||','||phone\_number||','||job\_id||','||hire\_date||','||salary||','||department\_id AS "the\_output" FROM EMPLOYEE;**

**Practice 2**

**1. Create a query to display the last name and salary of employees earning more than $12,000.  
 Place your SQL statement in a text file named lab2\_1.sql. Run your query.**

Answer : select last\_name,salary

from employees

where salary>12000;

**2. Create a query to display the employee last name and department number for employee number 176.**

Answer : select last\_name , department\_id

from employees

where employee\_id=176;

**3. Modify lab2\_1.sql to display the last name and salary for all employees whose salary is not in the range of $5,000 and $12,000. Place your SQL statement in a text file named lab2\_3.sq**

Answer : select last\_name,salary

from employees

where salary not between 5000 and 12000;

**4. Display the employee last name, job ID, and start date of employees hired between February 20, 1998, and May 1, 1998. Order the query in ascending order by start date.**

Answer : select last\_name,job\_id,hire\_date

from employees

where hire\_date between '20-feb-98' and '1-MAY-98';

**5. Display the last name and department number of all employees in departments 20 and 50 in alphabetical order by name.**

Answer : select last\_name ,department\_id

from employees

where department\_id IN (20 ,50);

**6. Modify lab2\_3.sql to list the last name and salary of employees who earn between $5,000 and $12,000, and are in department 20 or 50. Label the columns Employee and Monthly Salary, respectively. Resave lab2\_3.sql as lab2\_6.sql. Run the statement in lab2\_6.sql.**

Answer : select last\_name "Employee",salary "Monthly Salary"

from employees

where salary between 5000 and 12000

and department\_id in (20,50);

**7. Display the last name and hire date of every employee who was hired in 1994.**

Answer : select last\_name,hire\_date

from employees

where hire\_date ='07-june-94';

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

Answer : select last\_name ,job\_id

from employees

where job\_id not like 'IT\_PROG';

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

Answer : select last\_name,salary,commission\_pct

from employees

order by commission\_pct asc;

**10. Display the last names of all employees where the third letter of the name is an *a.***

Answer : select last\_name

from employees

where last\_name like '\_\_a%';

**11. Display the last name of all employees who have an *a* and an *e* in their last name**

Answer : select last\_name

from employees

where last\_name like '\_a%'

and last\_name like '\_\_\_\_\_e%';

**12. Display the last name, job, and salary for all employees whose job is sales representative or stock clerk and whose salary is not equal to $2,500, $3,500, or $7,000**

Answer : select last\_name,job\_id,salary

from employees

where job\_id IN('ST\_CLERK','SA\_REP')

AND SALARY not in('2500','3500','7000');

**13. Modify lab2\_6.sql to display the last name, salary, and commission for all employees whose commission amount is 20%. Resave lab2\_6.sql as lab2\_13.sql. Rerun the statement in lab2\_13.sql.**

Answer : select last\_name "Employee",salary "Monthly Salary",commission\_pct

from employees

where salary between 5000 and 12000

and department\_id in (20,50)

OR Commission\_pct in ('0.2');

**Practice 3 - Part One**

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

Answer : select SYSDATE from DUAL;

2. For each employee, display the employee number, last\_name, salary, and salary increased by 15% and expressed as a whole number. Label the column New Salary. Place your SQL statement in a text file named lab3\_2.sql.

Answer : select employee\_id,last\_name,salary,round(salary\*1.15) "New Salary"

from employees;

3. Run your query in the file lab3\_2.sql.

Answer : select \* from employees;

4. Modify your query lab3\_2.sql to add a column that subtracts the old salary from  
 the new salary. Label the column Increase. Save the contents of the file as lab3\_4.sql. Run the revised query.

Answer : select employee\_id,last\_name,salary,round(salary\*1.15)"New Salary",(salary\*1.15)-salary"increase"

from employees;

**Practice 3, Part One: Overview (continued)**

5. Write a query that displays the employee’s last names with the first letter capitalized and all other letters lowercase, and the length of the names, for all employees whose name starts with J, A, or M. Give each column an appropriate label. Sort the results by the employees’ last names.

Answer : select initcap(last\_name)"Name",length(last\_name)

from employees

where last\_name LIKE 'M%'

OR LAST\_NAME LIKE 'A%';

**Practice 3 - Part Two**

6. For each employee, display the employee’s last 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.

**Note:** Your results will differ.

Answer : SELECT last\_name,round(months\_between(Sysdate,hire\_date)/12)"Month work"

from employees;

**7. Write a query that produces the following for each employee:  
 <employee last name> earns <salary> monthly but wants <3 times salary>. Label the column Dream Salaries.**

Answer SELECT last\_name||' earn '|| salary ||' monthly but wants '||salary\*3

from employees;

**8. Create a query to display the last name and salary for all employees. Format the salary to be 15  
 characters long, left-padded with $. Label the column SALARY.**

Answer SELECT lpad(salary,15,'$') SALARY

FROM employees;

**Practice 3 - Part Two (continued)**

9. Display each employee’s last 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 “Monday, the Thirty-First of July, 2000.”

Answer : select last\_name, hire\_date,TO\_CHAR(next\_day(add\_month(hire\_date, 6), 'monday'),'fmday, Month DDth, YYYY')

from employees;

1. Display the last name, hire date, and day of the week on which the employee started. Label  
    the column DAY. Order the results by the day of the week starting with Monday.

Answer :

select last\_name , hire\_date,to\_char(hire\_date,'DAY') DAy

from employees

order by to\_char(hire\_date-1,'d');

1. Create a query that displays the employees’ last names and commission amounts. If an employee does not earn commission, put “No Commission.” Label the column COMM.

Answer :

select last\_name,NVL(to\_char(commission\_pct),'no commission')comm

from employees;

1. **Create a query that displays the employees’ last names and indicates the amounts of their annual salaries with asterisks. Each asterisk signifies a thousand dollars. Sort the data in descending order of salary. Label the column EMPLOYEES\_AND\_THEIR\_SALARIES.**

Answer :

**Practice 3 - Part Two (continued)**

13. Using the DECODE function, write a query that displays the grade of all employees based on the value of the column JOB\_ID, as per the following data:

***Job Grade***

AD\_PRES A

ST\_MAN B

IT\_PROG C

SA\_REP D

ST\_CLERK E

None of the above 0

Answer :

**Practice 4 - Part One**

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

Answer: select e.last\_name,e.department\_id,d.department\_name

from employees e, departments d;

**2. Create a unique listing of all jobs that are in department 80. Include the location of the department in the output.**

Answer: select distinct(e.job\_id),l.location\_id

from employees e, locations l

where e.department\_id =80;

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

Answer: select e.last\_name,d.department\_name,l.location\_id,city

from employees e, locations l,departments d;

**4. Display the employee last name and department name for all employees who have an *a* (lowercase) in their last names. Place your SQL statement in a text file named lab4\_4.sql.**

Answer: select e.last\_name,d.department\_name

from employees e,departments d

where e.last\_name like '\_\_\_a%';

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

Answer: SELECT e.last\_name,e.job\_id,e.department\_id,d.department\_name

FROM employees e, departments d,locations l

WHERE city like'%Toronto%' ;

**6. Display the employee last name and employee number along with their manager’s last name and manager number. Label the columns Employee, Emp#, Manager, and Mgr#, respectively.  
Place your SQL statement in a text file named lab4\_6.sql.**

Answer: SELECT worker.last\_name"Employee",worker.employee\_id"EMP#", manager.last\_name"Manager",manager.employee\_id"Mgr#"

FROM employees worker, employees manager

WHERE worker.manager\_id = manager.employee\_id

Below problem was from outer join read it

**7. Modify lab4\_6.sql to display all employees including King, who has no manager. Order the results by the employee number.  
Place your SQL statement in a text file named lab4\_7.sql. Run the query in lab4\_7.sql**

Answer: SELECT worker.last\_name"Employee",worker.employee\_id"EMP#", manager.last\_name"Manager",manager.employee\_id"Mgr#"

FROM employees worker, employees manager

WHERE worker.manager\_id (+)= manager.employee\_id ;

**Practice 5**

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

**1. Group functions work across many rows to produce one result per group.**

Answer: **True/False**

**2. Group functions include nulls in calculations.**

Answer: **True/False**

**3. The WHERE clause restricts rows prior to inclusion in a group calculation.**

Answer: **True/False**

**4. Display the highest, lowest, sum, and average salary of all employees. Label the columns  
 Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number. Place your SQL statement in a text file named lab5\_4.sql**

Answer: select MAX(SALARY)"Maximum",MIN(SALARY)"Minimun",SUM(SALARY)"SUM",AVG(SALARY)"Average"

from employees;

**5. Modify the query in lab5\_4.sql to display the minimum, maximum, sum, and average salary for each job type. Resave lab5\_4.sql to lab5\_5.sql. Run the statement in lab5\_5.sql.**

Answer: select job\_id,MAX(SALARY)"Maximum",MIN(SALARY)"Minimun",SUM(SALARY)"SUM",AVG(SALARY)"Average"

from employees

group by job\_id;

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

Answer: select job\_id,count(\*)

from employees

group by job\_id;

**7. Determine the number of managers without listing them. Label the column Number of  
 Managers. *Hint: Use the MANAGER\_ID column to determine the number of managers.***

Answer: select manager\_id "number of managers",COUNT(\*)

FROM EMPLOYEES

GROUP BY manager\_id;

**8. Write a query that displays the difference between the highest and lowest salaries. Label the column DIFFERENCE.**

Answer: select max(salary)-min(salary)

FROM EMPLOYEES;

**9. Display the manager number and the salary of the lowest paid employee for that manager.  
 Exclude anyone whose manager is not known. Exclude any groups where the minimum  
 salary is $6,000 or less. Sort the output in descending order of salary.**

Answer: select manager\_id , max(salary)

FROM EMPLOYEES

group by manager\_id

having min(salary)<=6000;

**10. Write a query to display each department’s name, location, number of employees, and the average salary for all employees in that**

**department. Label the columns Name, Location, Number of People, and Salary, respectively. Round the average salary to two decimal places.**

Answer:

**11. Create a query that will display the total number of employees and, of that total, the number of employees hired in 1995, 1996, 1997, and 1998. Create appropriate column headings.**

Answer:

**12. Create a matrix query to display the job, the salary for that job based on department number, and the total salary for that job, for departments 20, 50, 80, and 90, giving each column an appropriate heading.**

Answer:

Chapter 8

**Insert data into the MY\_EMPLOYEE table.**

1. **Run the statement in the lab8\_1.sql script to build the MY\_EMPLOYEE table to be used for the lab.**

Answer : create table my\_employees(ID number(30),LAST\_NAME varchar(25),FIRST\_NAME varchar(25),UNSERID varchar(9),SALARY NUMBER(9,2));

1. **Describe the structure of the MY\_EMPLOYEE table to identify the column names.**

Answer: describe my\_employees;

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

Answer: insert into my\_employees(id,last\_name,first\_name,unserid,salary)

values(1,'Patel','Ralph','rpatal',895);

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

Answer: insert into my\_employees(id,last\_name,first\_name,unserid,salary)

values(2,'Dancs','Betty','bdancs',860);

**5. Confirm your addition to the table**

Answer: insert into my\_employees(id,last\_name,first\_name,unserid,salary)

values(2,'Dancs','Betty','bdancs',860);

**6. Write an insert statement in a text file named loademp.sql to load rows into the MY\_EMPLOYEE table. Concatenate the first letter of the first name and the first seven characters of the last name to produce the user ID.**

**7. Populate the table with the next two rows of sample data by running the insert statement in the script that you created.**

Answer 1: insert into my\_employees(id,last\_name,first\_name,unserid,salary)

values(3,'Biri','Ben','bbiri',1100);

Answer 2: insert into my\_employees(id,last\_name,first\_name,unserid,salary)

values(4,'Newman','Chad','cnewman',750);

**8. Confirm your additions to the table.**

Answer: select\*from my\_employees;

**9. Make the data additions permanent.**

Answer:

**Update and delete data in the MY\_EMPLOYEE table.**

**10. Change the last name of employee 3 to Drexler.**

Answer: UPDATE MY\_EMPLOYEES

set last\_name='Drexler'

where id=3;

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

Answer: UPDATE MY\_EMPLOYEES

set salary=1000

where salary<=1000;

**12. Verify your changes to the table.**

Answer: select\*from my\_employees;

**13. Delete Betty Dancs from the MY\_EMPLOYEES table.**

Answer: delete

from my\_employees

where first\_name='Betty';

**14. Confirm your changes to the table.**

Answer: select \* from my\_employees;

**15. Commit all pending changes.**

Answer:

**Control data transaction to the MY\_EMPLOYEE table.**

**16. Populate the table with the last row of sample data by modifying the statements in the script that you created in step 6. Run the statements in the script.**

Answer:

**17. Confirm your addition to the table.**

Answer:

**18. Mark an intermediate point in the processing of the transaction.**

Answer:

**19. Empty the entire table.**

Answer:

**20. Confirm that the table is empty.**

Answer:

**21. Discard the most recent DELETE operation without discarding the earlier INSERT operation.**

Answer:

**22. Confirm that the new row is still intact**

Answer:

**23. Make the data addition permanent.**

Answer:

**Practice 9**

**1. Create the DEPT table based on the following table instance chart. Place the  
syntax in a script called lab9\_1.sql, then execute the statement in the script to create the table. Confirm that the table is created.**

****

Answer: create table depts(id number(9),name varchar2(25));

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

Answer: insert into depts

select department\_id,department\_name

from departments;

**3. Create the EMP table based on the following table instance chart. Place the syntax in  
 a script called lab9\_3.sql, and then execute the statement in the script to create the table. Confirm that the table is created.**

****

Answer: create table emp(id number(7),last\_name varchar2(25),first\_name varchar2(25),DEPT number(7) );

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

Answer: alter table emp

modify (last\_name varchar2(50));

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

Answer: SELECT \*

FROM CAT;

**6. Create the EMPLOYEES2 table based on the structure of the EMPLOYEES table. Include only the EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, SALARY, and DEPARTMENT\_ID columns. Name the columns in your new table ID, FIRST\_NAME, LAST\_NAME, SALARY , and DEPT\_ID, respectively.**

Answer1: create table EMPLOYEE2(id number(4),last\_name varchar2(25),first\_name varchar2(25),SALARY NUMBER(8,2), department\_id varchar2(25));

Answer2: rename dept to detail\_dept;

Answer3: rename employee\_id to detail\_id;

**7. Drop the EMP table.**

Answer: drop table emp;

**8. Rename the EMPLOYEES2 table as EMP.**

Answer: rename employee2 to detail\_emp;

**9. Add a comment to the DEPT and EMP table definitions describing the tables. Confirm your additions in the data dictionary.**

Answer:

**10. Drop the FIRST\_NAME column from the EMP table. Confirm your modification by checking the description of the table.**

Answer: alter table emp

drop column last\_name;

**11. In the EMP table, mark the DEPT\_ID column in the EMP table as UNUSED. Confirm your modification by checking the description of the table.**

Answer: alter table emp

set unused (dept\_id);

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

Answer: alter table emp

drop unused columns;

**Practice 10**

1. Add a table-level PRIMARY KEY constraint to the EMP table on the ID column. The constraint should be named at creation. Name the constraint my\_emp\_id\_pk.

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

successfully.

2. Create a PRIMARY KEY constraint to the DEPT table using the ID column. The constraint should be named at creation. Name the constraint my\_dept\_id\_pk.

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

successfully.

3. Add a column DEPT\_ID to the EMP table. Add a foreign key reference on the EMP table that ensures that the employee is not assigned to a nonexistent department. Name the constraint my\_emp\_dept\_id\_fk.

4. Confirm that the constraints were added by querying the USER\_CONSTRAINTS view. Note the types and names of the constraints. Save your statement text in a file called lab10\_4.sql.

5. Display the object names and types from the USER\_OBJECTS data dictionary view for the EMP and DEPT tables. Notice that the new tables and a new index were created.

If you have time, complete the following exercise:

6. Modify the EMP table. Add a COMMISSION column of NUMBER data type, precision 2, scale 2. Add a constraint to the commission column that ensures that a commission value is greater than zero.