Additional Practice Solutions

Additional Practices Solutions

The following exercises can be used for extra practice after you have discussed the data manipulation language (DML) and data definition language (DDL) statements in the lessons titled “Managing Schema Objects” and “Manipulating Large Data Sets.”

Note: Run the `lab_ap_cre_special_sal.sql`, `lab_ap_cre_sal_history.sql`, and `lab_ap_cre_mgr_history.sql` scripts in the labs folder to create the `SPECIAL_SAL`, `SAL_HISTORY`, and `MGR_HISTORY` tables.

1. The Human Resources department wants a list of underpaid employees, the salary history of employees, and the salary history of managers based on an industry salary survey. So they have asked you to do the following:

   Write a statement to do the following:

   - Retrieve the employee ID, hire date, salary, and manager ID of those employees whose employee ID is more than or equal to 200 from the `EMPLOYEES` table.
   - If the salary is less than $5,000, insert the employee ID and salary into the `SPECIAL_SAL` table.
   - Insert the employee ID, hire date, and salary into the `SAL_HISTORY` table.
   - Insert the employee ID, manager ID, and salary into the `MGR_HISTORY` table.

```
INSERT ALL
WHEN SAL < 5000 THEN
INTO special_sal VALUES (EMPID, SAL)
ELSE
INTO sal_history VALUES (EMPID,HIREDATE,SAL)
INTO mgr_history VALUES (EMPID,MGR,SAL)
SELECT employee_id EMPID, hire_date HIREDATE,
    salary SAL, manager_id MGR
FROM employees
WHERE employee_id >=200;
```

2. Query the `SPECIAL_SAL`, `SAL_HISTORY`, and `MGR_HISTORY` tables to view the inserted records.

```
SELECT * FROM special_sal;
SELECT * FROM sal_history;
SELECT * FROM mgr_history;
```

Additional Practices Solutions (continued)

3. The DBA wants you to create a table, which has a primary key constraint, but the DBA wants the index to have a different name than the constraint. Create the LOCATIONS_NAMED_INDEX table based on the following table instance chart.

Column Name	Deptno	Dname
Primary Key	Yes	
Data Type	Number	VARCHAR2
Length	4	30

Name the index for the PRIMARY KEY column as LOCATIONS_PK_IDX.

```
CREATE TABLE LOCATIONS_NAMED_INDEX
(location_id NUMBER(4) PRIMARY KEY USING INDEX
(CREATE INDEX locations_pk_idx ON
LOCATIONS_NAMED_INDEX(location_id)),
location_name VARCHAR2(20));
```

4. Query the USER_INDEXES table to display INDEX_NAME for the LOCATIONS_NAMED_INDEX table.

```
SELECT INDEX_NAME, TABLE_NAME
FROM USER_INDEXES
WHERE TABLE_NAME = 'LOCATIONS_NAMED_INDEX';
```

The following exercises can be used for extra practice after you have discussed enhancements to the GROUP BY clause.

5. The Human Resources department requires some reports on certain departments. These are its requirements:

   Write a query to display the following for those departments whose department ID is greater than 80:

   - The total salary for every job within a department
   - The total salary
   - The total salary for those cities in which the departments are located
   - The total salary for every job, irrespective of the department
   - The total salary for every department irrespective of the city
   - The total salary for the departments, irrespective of the job titles and cities

```
COLUMN     city FORMAT A25 Heading CITY
COLUMN     department_name FORMAT A15 Heading DNAME
COLUMN     job_id FORMAT A10 Heading JOB
COLUMN     SUM(salary) FORMAT $99,99,999.00 Heading
           SUM(SALARY)

SELECT     l.city, d.department_name, e.job_id,
           SUM(e.salary)
FROM       locations l, employees e, departments d
WHERE      d.location_id = l.location_id
AND        e.department_id = d.department_id
AND        e.department_id > 80
GROUP BY CUBE( l.city, d.department_name, e.job_id);
```

6. The Accounting department requires an analysis on the maximum and minimum salaries by department, job, and manager. They have asked you to do the following:

   Write a query to display the following groupings:

   - Department ID, Job ID
   - Job ID, Manager ID

   The query should calculate the maximum and minimum salaries for each of these groups.

```
SELECT
  department_id,job_id,manager_id,max(salary),
    min(salary)
FROM   employees
GROUP BY GROUPING SETS
       ((department_id,job_id), (job_id,manager_id));
```

The following exercises can be used for extra practice after you have discussed the datetime functions.

You work for a global company and the new vice president of operations wants to know the different time zones of all the company branches. He has requested the following information:

7. Alter the session to set the NLS_DATE_FORMAT to DD-MON-YYYY HH24:MI:SS.

```
ALTER SESSION
SET NLS_DATE_FORMAT = ‘DD-MON-YYYY HH24:MI:SS’;
```

8. a. Write queries to display the time zone offsets (TZ_OFFSET) for the following time zones:

   - Australia/Sydney

```
SELECT TZ_OFFSET (‘Australia/Sydney’) from dual;
```

   - Chile/Easter Island

```
SELECT TZ_OFFSET (‘Chile/EasterIsland’) from dual;
```

b. Alter the session to set the TIME_ZONE parameter value to the time zone offset of Australia/Sydney.

```
ALTER SESSION SET TIME_ZONE = ‘+10:00’;
```

c. Display SYSDATE, CURRENT_DATE, CURRENT_TIMESTAMP, and LOCALTIMESTAMP for this session.
Note: The output may be different based on the date when the command is executed.

```
SELECT SYSDATE, CURRENT_DATE, CURRENT_TIMESTAMP,
LOCALTIMESTAMP FROM DUAL;
```

d. Alter the session to set the TIME_ZONE parameter value to the time zone offset of Chile/Easter Island.

Note: The results of the preceding question are based on a different date, and in some cases, they will not match the actual results that the students get. Also, the time zone offset of the various countries may differ based on the daylight saving time.

```
ALTER SESSION SET TIME_ZONE = ‘-06:00’;
```

e. Display SYSDATE, CURRENT_DATE, CURRENT_TIMESTAMP, and LOCALTIMESTAMP for this session.

Note: The output may be different based on the date when the command is executed.

```
SELECT SYSDATE, CURRENT_DATE, CURRENT_TIMESTAMP,
LOCALTIMESTAMP FROM DUAL;
```

f. Alter the session to set NLS_DATE_FORMAT to DD-MON-YYYY.

```
ALTER SESSION SET NLS_DATE_FORMAT = ‘DD-MON-YYYY’;
```

Additional Practices Solutions (continued)

Note

- Observe in the preceding question that CURRENT_DATE, CURRENT_TIMESTAMP, and LOCALTIMESTAMP are all sensitive to the session time zone. Observe that SYSDATE is not sensitive to the session time zone.
- The results of the preceding question are based on a different date, and in some cases, they will not match the actual results that the students get. Also, the time zone offset of the various countries may differ based on the daylight saving time.

9. The Human Resources department wants a list of employees who are up for review in January, so they have requested you to do the following:

   Write a query to display the last names, month of the date of hire, and hire date of those employees who have been hired in the month of January, irrespective of the year of hire.

```
SELECT last_name, EXTRACT (MONTH FROM HIRE_DATE),
HIRE_DATE FROM employees
WHERE EXTRACT (MONTH  FROM  HIRE_DATE) = 1;
```

The following exercises can be used for extra practice after you have discussed advanced subqueries.

10. The CEO needs a report on the top three earners in the company for profit sharing. He has asked you to provide him with a list.

    Write a query to display the top three earners in the EMPLOYEES table. Display their last names and salaries.

```
SELECT last_name, salary
FROM  employees e
WHERE 3  > (SELECT COUNT  (*)
FROM  employees
WHERE e.salary < salary);
```

11. The benefits for the state of California have been changed based on a local ordinance. So the benefits representative has asked you to compile a list of people who are affected. Write a query to display the employee ID and last name of the employees who work in the state of California.
    Hint: Use scalar subqueries.

```
SELECT employee_id, last_name
  FROM employees e
  WHERE ((SELECT location_id
         FROM departments d
        WHERE e.department_id = d.department_id )
             IN  (SELECT location_id
                  FROM locations l
                  WHERE state_province = 'California'));
```

12. The DBA wants to remove old information from the database. One of the things that the DBA thinks is unnecessary is the old employment records. He or she has asked you to do the following:

    Write a query to delete the oldest `JOB_HISTORY` row of an employee by looking up the `JOB_HISTORY` table for `MIN(START_DATE)` for the employee. Delete the records of *only* those employees who have changed at least two jobs.
    Hint: Use a correlated `DELETE` command.

```
 DELETE FROM job_history JH
  WHERE employee_id = (SELECT employee_id
             FROM employees E
             WHERE JH.employee_id = E.employee_id
             AND START_DATE = (SELECT MIN(start_date)
                  FROM job_history  JH
                    WHERE JH.employee_id = E.employee_id)
                  AND 3 > (SELECT COUNT(*)
                  FROM job_history JH
                    WHERE JH.employee_id = E.employee_id
                    GROUP BY EMPLOYEE_ID
                    HAVING COUNT(*) >=  2));
```

13. The vice president of Human Resources needs the complete employment records for his annual employee recognition banquet speech. He makes a quick phone call to stop you from following the DBA's orders.

    Roll back the transaction.

```
        ROLLBACK;
```

14. The sluggish economy is forcing the management to take cost reduction actions. The CEO wants to review the highest paid jobs in the company. He has requested a list from you based on the following specifications:

    Write a query to display the job IDs of those jobs whose maximum salary is above half the maximum salary in the entire company. Use the `WITH` clause to write this query. Name the query `MAX_SAL_CALC`.

```
WITH
MAX_SAL_CALC AS (SELECT job_title, MAX(salary) AS
job_total
FROM employees, jobs
WHERE employees.job_id = jobs.job_id
GROUP BY job_title)
SELECT job_title, job_total
FROM MAX_SAL_CALC
WHERE job_total > (SELECT MAX(job_total) * 1/2
FROM MAX_SAL_CALC)
ORDER BY job_total DESC;
```

The following exercises can be used for extra practice after you have discussed hierarchical retrieval.

15. Lex De Haan is quitting the company. His replacement wants reports of his direct reports.

    Write a SQL statement to display the employee number, last name, start date, and salary, showing:

    a. De Haan’s direct reports:

```
SELECT employee_id, last_name, hire_date, salary
FROM employees
WHERE manager_id = (SELECT employee_id
FROM employees
WHERE last_name = ‘De Haan’);
```

    b. The organization tree under De Haan (employee number 102):

```
SELECT employee_id, last_name, hire_date, salary
FROM employees
WHERE employee_id != 102
CONNECT BY manager_id = PRIOR employee_id
START WITH employee_id = 102;
```

Additional Practices Solutions (continued)

16. Write a hierarchical query to display the employee number, manager number, and employee last name for all employees who are two levels below employee De Haan (employee number 102). Also display the level of the employee.

```
SELECT  employee_id, manager_id, level, last_name
 FROM employees
 WHERE LEVEL = 3
 CONNECT BY manager_id = PRIOR employee_id
 START WITH employee_id = 102;
```

17. The CEO wants a hierarchical report on all employees. He has given you the following requirements:

    Produce a hierarchical report to display the employee number, manager number, the `LEVEL` pseudocolumn, and employee last name. For every row in the `EMPLOYEES` table, you should print a tree structure that shows the employee, the employee's manager, the manager's manager, and so on. Use indentations for the `NAME` column.

```
COLUMN name FORMAT A25
SELECT  employee_id, manager_id, LEVEL,
LPAD(last_name, LENGTH(last_name)+(LEVEL*2)-2,'_')
   LAST_NAME
FROM  employees
CONNECT BY employee_id = PRIOR manager_id;
COLUMN name CLEAR
```