Oracle Database 10*g*: SQL Fundamentals II

Additional Practices and Solutions

D17111GC21

Edition 2.1

December 2006

D48223

ORACLE

Authors

Chaitanya Koratamaddi
Priya Vennapusa

Technical Contributors and Reviewers

Claire Bennett
Brian Boxx
Zarko Cesljas
Laurent Dereac
Nancy Greenberg
Angelika Krupp
Malika Marghadi
Priya Nathan
Bryan Roberts
Lata Shivaprasad
Naoko Susuki
Narayanan Radhakrishnan

Editors

Nita Pavitran
Atanu Raychaudhuri

Graphic Designer

Sanjeev Sharma

Publisher

Jobi Varghese

Contents

Additional Practices

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

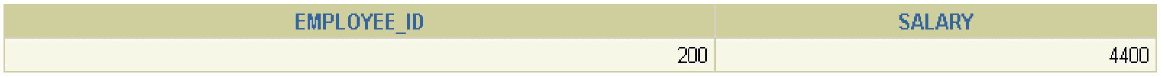
1. The Human Resources department wants to get a list of underpaid employees, salary history of employees, and salary history of managers based on an industry salary survey. So they have asked you to do the following:

   Write a statement to do the following:

   - Retrieve the details of 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 details of employee ID and salary into the `SPECIAL_SAL` table.
   - Insert the details of employee ID, hire date, and salary into the `SAL_HISTORY` table.
   - Insert the details of employee ID, manager ID, and salary into the `MGR_HISTORY` table.

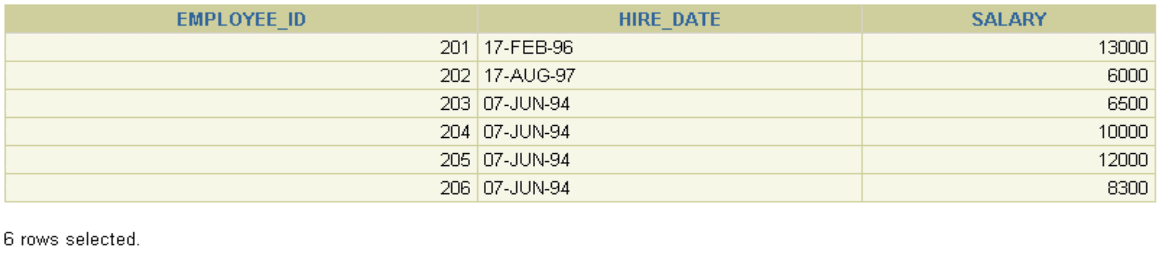
2. Query the `SPECIAL_SAL`, `SAL_HISTORY` and `MGR_HISTORY` tables to view the inserted records.

`SPECIAL_SAL`



EMPLOYEE_ID	SALARY
200	4400

`SALARY_HISTORY`



EMPLOYEE_ID	HIRE_DATE	SALARY
201	17-FEB-96	13000
202	17-AUG-97	6000
203	07-JUN-94	6500
204	07-JUN-94	10000
205	07-JUN-94	12000
206	07-JUN-94	8300

6 rows selected.

MGR_HISTORY

EMPLOYEE_ID	MANAGER_ID	SALARY
201	100	13000
202	201	6000
203	101	6500
204	101	10000
205	101	12000
206	205	8300

```
6 rows selected.
```

3. The DBA needs you to create a table, which has a primary key constraint, but she wants to name the index to have a different name than the constraint. Create the `LOCATIONS_NAMED_INDEX` table based on the following table instance chart. Name the index for the `PRIMARY KEY` column as `LOCATIONS_PK_IDX`.

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

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

INDEX_NAME	TABLE_NAME
LOCATIONS_PK_IDX	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 their 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 job titles and cities

CITY	DNAME	JOB	SUM(E.SALARY)
			129900
		AD_VP	34000
		AC_MGR	12000
		FI_MGR	12000
		AD_PRES	24000
		AC_ACCOUNT	8300
		FI_ACCOUNT	39600
	Finance		51600
	Finance	FI_MGR	12000
	Finance	FI_ACCOUNT	39600
	Executive		58000

. . .

CITY	DNAME	JOB	SUM(E.SALARY)
Seattle	Finance	FI_MGR	12000
Seattle	Finance	FI_ACCOUNT	39600
Seattle	Executive		58000
Seattle	Executive	AD_VP	34000
Seattle	Executive	AD_PRES	24000
Seattle	Accounting		20300
Seattle	Accounting	AC_MGR	12000
Seattle	Accounting	AC_ACCOUNT	8300

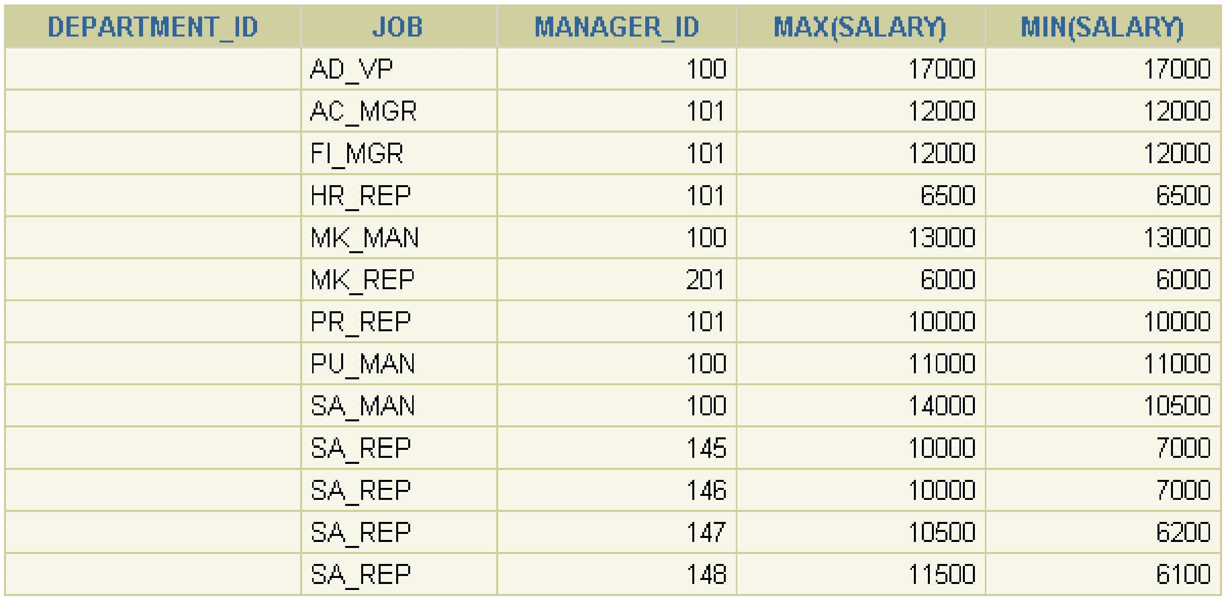
32 rows selected.

6. The Accounting department requires an analysis on 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:

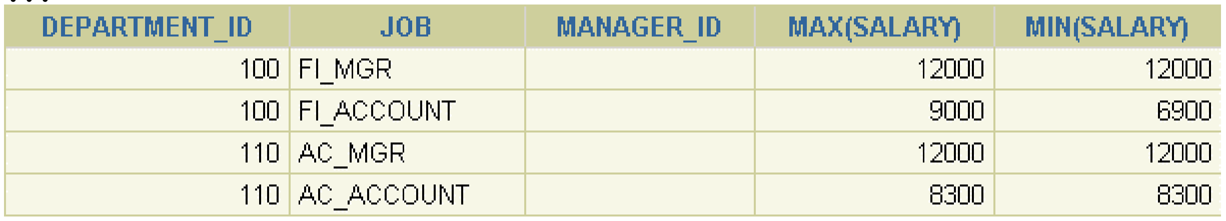
   - Department ID, Job ID
   - Job ID, Manager ID

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



DEPARTMENT_ID	JOB	MANAGER_ID	MAX(SALARY)	MIN(SALARY)
	AD_VP	100	17000	17000
	AC_MGR	101	12000	12000
	FI_MGR	101	12000	12000
	HR_REP	101	6500	6500
	MK_MAN	100	13000	13000
	MK_REP	201	6000	6000
	PR_REP	101	10000	10000
	PU_MAN	100	11000	11000
	SA_MAN	100	14000	10500
	SA_REP	145	10000	7000
	SA_REP	146	10000	7000
	SA_REP	147	10500	6200
	SA_REP	148	11500	6100

. . .



DEPARTMENT_ID	JOB	MANAGER_ID	MAX(SALARY)	MIN(SALARY)
100	FI_MGR		12000	12000
100	FI_ACCOUNT		9000	6900
110	AC_MGR		12000	12000
110	AC_ACCOUNT		8300	8300

52 rows selected.

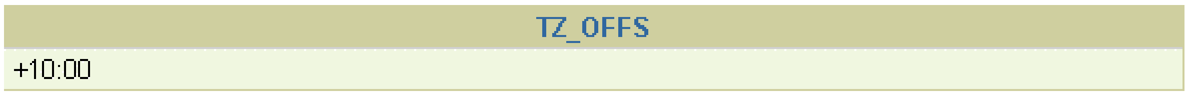
The following exercises can be used for extra practice after you have discussed datetime functions.

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`.

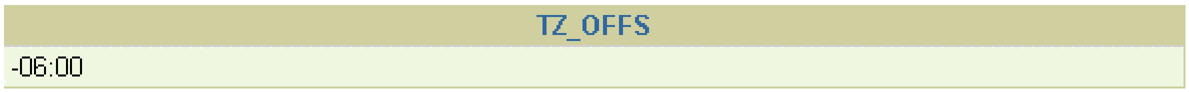
8. a. Write queries to display the time zone offsets (`TZ_OFFSET`) for the following time zones:

   Australia/Sydney



TZ_OFFS
+10:00

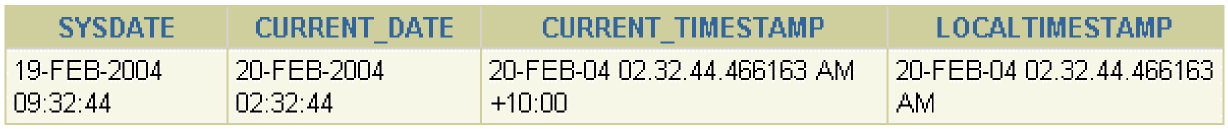
   Chile/Easter Island



TZ_OFFS
-06:00

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

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



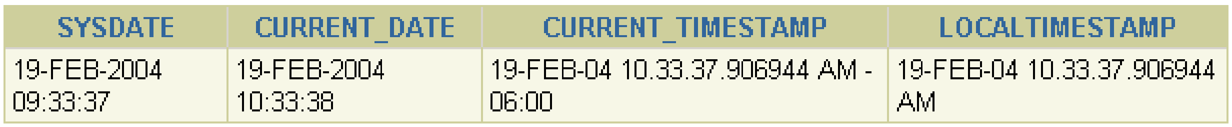
SYSDATE	CURRENT_DATE	CURRENT_TIMESTAMP	LOCALTIMESTAMP
19-FEB-2004 09:32:44	20-FEB-2004 02:32:44	20-FEB-04 02.32.44.466163 AM +10:00	20-FEB-04 02.32.44.466163 AM

   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 daylight saving time.

   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.



SYSDATE	CURRENT_DATE	CURRENT_TIMESTAMP	LOCALTIMESTAMP
19-FEB-2004 09:33:37	19-FEB-2004 10:33:38	19-FEB-04 10.33.37.906944 AM -06:00	19-FEB-04 10.33.37.906944 AM

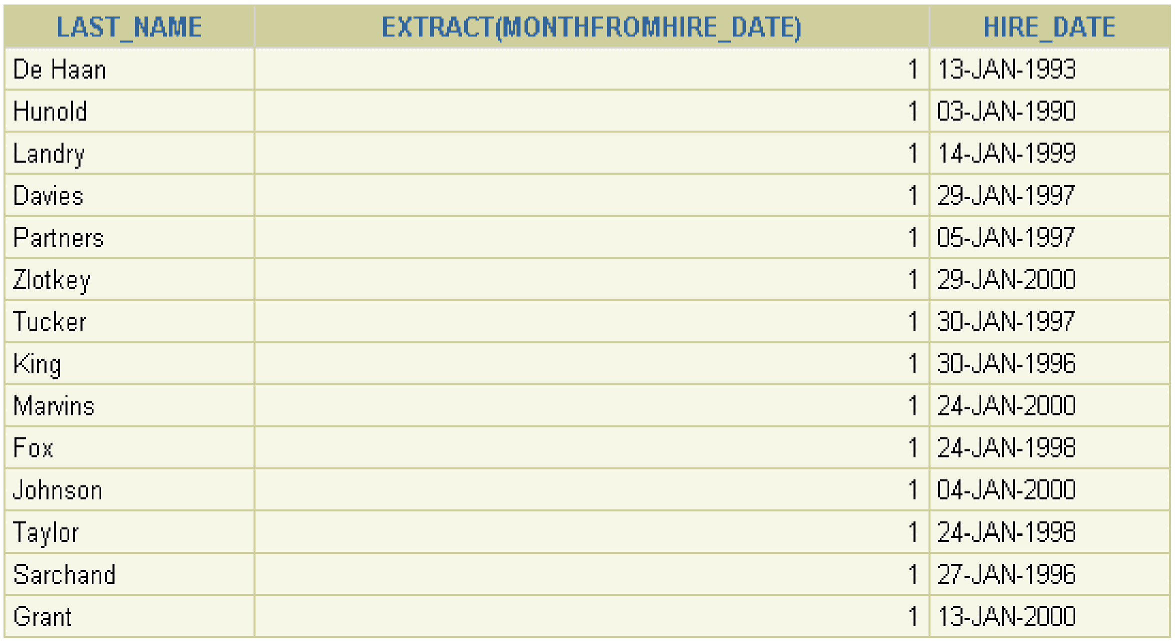
   f. Alter the session to set the `NLS_DATE_FORMAT` to `DD-MON-YYYY`.

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 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.



LAST_NAME	EXTRACT(MONTHFROMHIRE_DATE)	HIRE_DATE
De Haan	1	13-JAN-1993
Hunold	1	03-JAN-1990
Landry	1	14-JAN-1999
Davies	1	29-JAN-1997
Partners	1	05-JAN-1997
Zlotkey	1	29-JAN-2000
Tucker	1	30-JAN-1997
King	1	30-JAN-1996
Marvins	1	24-JAN-2000
Fox	1	24-JAN-1998
Johnson	1	04-JAN-2000
Taylor	1	24-JAN-1998
Sarchand	1	27-JAN-1996
Grant	1	13-JAN-2000

14 rows selected.

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.

LAST_NAME	SALARY
King	24000
Kochhar	17000
De Haan	17000

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 the people who are affected. Write a query to display the employee ID and last names of the employees who work in the state of California.
    Hint: Use scalar subqueries.

EMPLOYEE_ID	LAST_NAME
120	Weiss
121	Fripp
122	Kaufling
123	Vollman
124	Mourgos
125	Nayer
126	Mikkilineni
127	Landry
128	Markle
129	Bissot
. . .	
190	Gates
191	Perkins
192	Bell
193	Everett
194	McCain
195	Jones
196	Walsh
197	Feeney
198	OConnell
199	Grant

```
45 rows selected.
```

12. The DBA wants to remove old information from the database. One of the things she thinks is unnecessary is the old employment records. 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 the `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.

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.

14. The sluggish economy is forcing 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`.

JOB_TITLE	JOB_TOTAL
President	24000
Administration Vice President	17000
Sales Manager	14000
Marketing Manager	13000

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 employee number, last name, start date, and salary, showing:

    a. De Haan's direct reports:

EMPLOYEE_ID	LAST_NAME	HIRE_DATE	SALARY
103	Hunold	03-JAN-1990	9000

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

EMPLOYEE_ID	LAST_NAME	HIRE_DATE	SALARY
103	Hunold	03-JAN-1990	9000
104	Ernst	21-MAY-1991	6000
105	Austin	25-JUN-1997	4800
106	Pataballa	05-FEB-1998	4800
107	Lorentz	07-FEB-1999	4200

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.

EMPLOYEE_ID	MANAGER_ID	LEVEL	LAST_NAME
104	103	3	Ernst
105	103	3	Austin
106	103	3	Pataballa
107	103	3	Lorentz

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 showing the employee, the employee's manager, then the manager's manager, and so on. Use indentations for the `NAME` column.

EMPLOYEE_ID	MANAGER_ID	LEVEL	LAST_NAME
100		1	King
101	100	1	Kochhar
100		2	__King
102	100	1	De Haan
100		2	__King
103	102	1	Hunold
102	100	2	__De Haan
100		3	____King

. . .

EMPLOYEE_ID	MANAGER_ID	LEVEL	LAST_NAME
205	101	2	__Higgins
101	100	3	____Kochhar
100		4	______King

315 rows selected.

Note: The output shown is only a sample. All the rows from the actual output are not included here.

Additional Practice Solutions

The following exercises can be used for extra practice after you have discussed 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 to get a list of underpaid employees, salary history of employees, and 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 details of 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 details of employee ID and salary into the `SPECIAL_SAL` table.
   - Insert the details of employee ID, hire date, and salary into the `SAL_HISTORY` table.
   - Insert the details of 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 the `MGR_HISTORY` tables to view the inserted records.

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

3. The DBA needs you to create a table, which has a primary key constraint, but she wants to name the index to have a different name than the constraint. Create the

LOCATIONS_NAMED_INDEX table based on the following table instance chart. Name the index for the PRIMARY KEY column as LOCATIONS_PK_IDX.

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

```
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 the 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 their 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 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 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 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 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 the `NLS_DATE_FORMAT` to `DD-MON-YYYY`.

```
ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MON-YYYY';
```

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 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 the people who are affected. Write a query to display the employee ID and last names 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 she thinks is unnecessary is the old employment records. 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 the `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 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 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;
```

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 showing the employee, the employee's manager, then 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
```