

2200290120001

Aaditiya Tyagi

CS5A

DBMS Lab Assessment :

1. Display name, job id and salary and a column revised salary where IT programmer gets a hike of 10%, clerk gets 15% hike and sales representative gets 20% hike and remaining get the same salary.

The screenshot shows the SQL Workshop interface with a query editor and a results table. The query is as follows:

```
SELECT
  first_name || ' ' || last_name AS name,
  job_id,
  salary,
  CASE
    WHEN job_id = 'IT_PROG' THEN salary * 1.10
    WHEN job_id = 'ST_CLERK' THEN salary * 1.15
    WHEN job_id = 'SA_REP' THEN salary * 1.20
    ELSE salary
  END AS revised_salary
FROM employees;
```

The results table displays the following data:

NAME	JOB_ID	SALARY	REVISED_SALARY
Steven King	AD_PRES	24000	24000
Neena Kochhar	AD_VP	17000	17000
Lex De Haan	AD_VP	17000	17000
Alexander Hunold	IT_PROG	9000	9900
Bruce Ernst	IT_PROG	6000	6600
David Austin	IT_PROG	4800	5280
Valli Pataballa	IT_PROG	4800	5280
Diana Lorentz	IT_PROG	4200	4620
Nancy Greenberg	FI_MGR	12008	12008
Daniel Faviet	FI_ACCOUNT	9000	9000

More than 10 rows available. Increase rows selector to view more rows.
10 rows returned in 0.00 seconds [Download](#)

2.) Using decode function write a query that display the grade of all the employees based on the column job_id as per the following data :

The screenshot shows an SQL IDE interface. At the top, there are tabs for 'Home', 'Application Builder', 'SQL Workshop', 'Team Development', and 'Administration'. Below these, a breadcrumb trail shows 'Home > SQL Workshop > SQL Commands'. The main area contains a query editor with the following SQL code:

```

SELECT
  first_name || ' ' || last_name AS name,
  job_id,
  DECODE(
    job_id,
    'AD_PRES', 'A',
    'ST_MAN', 'B',
    'IT_PROG', 'C',
    'SA_REP', 'D',
    'ST_CLERK', 'E',
    'NONE', '0',
    NULL
  ) AS grade
FROM employees;

```

Below the query editor, there are buttons for 'Autocommit' (checked), 'Rows' (set to 10), 'Save', and 'Run'. The 'Run' button is highlighted in yellow. Below the query editor, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is selected, showing a table with the following data:

NAME	JOB_ID	GRADE
William Gietz	AC_ACCOUNT	-
Shelley Higgins	AC_MGR	-
Jennifer Whalen	AD_ASST	-
Steven King	AD_PRES	A
Neena Kochhar	AD_VP	-
Lex De Haan	AD_VP	-
Daniel Faviot	FI_ACCOUNT	-
John Chen	FI_ACCOUNT	-
Ismael Sciarra	FI_ACCOUNT	-
Jose Manuel Urman	FI_ACCOUNT	-

Below the table, there is a message: 'More than 10 rows available. Increase rows selector to view more rows.' At the bottom, it says '10 rows returned in 0.01 seconds' and there is a 'Download' link.

3.) Display department name along with employee id of all the employees.

The screenshot shows the SQL Workshop interface. The top navigation bar includes 'Home', 'Application Builder', 'SQL Workshop', 'Team Development', and 'Administration'. The breadcrumb trail is 'Home > SQL Workshop > SQL Commands'. Below the breadcrumb, there is a toolbar with 'Autocommit' (checked), 'Rows' (set to 10), a pencil icon, an eraser icon, a 'Save' button, and a 'Run' button. The SQL editor contains the following query:

```
SELECT
  d.department_name,
  e.employee_id
FROM employees e
JOIN departments d ON e.department_id = d.department_id;
```

Below the editor, the 'Results' tab is selected, showing a table with two columns: 'DEPARTMENT_NAME' and 'EMPLOYEE_ID'. The table contains 10 rows of data. At the bottom of the results, it states 'More than 10 rows available. Increase rows selector to view more rows.' and '10 rows returned in 0.01 seconds' with a 'Download' link.

DEPARTMENT_NAME	EMPLOYEE_ID
Executive	100
Executive	101
Executive	102
IT	103
IT	104
IT	105
IT	106
IT	107
Finance	108
Finance	109

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.01 seconds [Download](#)



4.) Display name, job, salary for all the employees where job is sales representatives or clerk and whose salary is not equal to 2500, 3500, 7000.

HomeApplication BuilderSQL WorkshopTeam DevelopmentAdministration

HomeSQL WorkshopSQL CommandsScr

☒ Autocommit

Rows10



SaveRun

```
SELECT
  first_name || ' ' || last_name AS name,
  job_id,
  salary
FROM employees
WHERE job_id IN ('SA_REP', 'ST_CLERK')
AND salary NOT IN (2500, 3500, 7000);
```

ResultsExplainDescribeSaved SQLHistory

DEPARTMENT_NAME	EMPLOYEE_ID
Executive	100
Executive	101
Executive	102
IT	103
IT	104
IT	105
IT	106
IT	107
Finance	108
Finance	109

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.01 secondsDownload

[illegible]

HomeApplication BuilderSQL WorkshopTeam DevelopmentAdministration

HomeSQL WorkshopSQL CommandsSc

☒ AutocommitRows10SaveRun

```
SELECT
  job_id,
  SUM(salary) AS total_salary
FROM employees
WHERE job_id != 'SA_REP'
GROUP BY job_id
HAVING SUM(salary) > 10000
ORDER BY total_salary ASC;
```

ResultsExplainDescribeSaved SQLHistory

JOB_ID	TOTAL_SALARY
PU_MAN	11000
AC_MGR	12008
FI_MGR	12008
MK_MAN	13000
PU_CLERK	13900
AD_PRES	24000
IT_PROG	28800
AD_VP	34000
ST_MAN	36400
FI_ACCOUNT	39600

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.00 secondsDownload

7. Display department id, sum of salary jobwise in a department.

The screenshot shows the SQL Workshop interface. The top navigation bar includes 'Home', 'Application Builder', 'SQL Workshop', 'Team Development', and 'Administration'. The breadcrumb trail is 'Home > SQL Workshop > SQL Commands'. Below the navigation bar, there is a toolbar with 'Autocommit' (checked), 'Rows' (set to 10), a pencil icon, an eraser icon, a 'Save' button, and a 'Run' button. The main text area contains the following SQL query:

```
SELECT
  department_id,
  job_id,
  SUM(salary) AS job_salary
FROM employees
GROUP BY department_id, job_id;
```

Below the query editor, there is a tabbed interface with 'Results' selected. The results are displayed in a table with the following columns: DEPARTMENT_ID, JOB_ID, and JOB_SALARY.

DEPARTMENT_ID	JOB_ID	JOB_SALARY
110	AC_ACCOUNT	8300
90	AD_VP	34000
50	ST_CLERK	55700
80	SA_REP	243500
50	ST_MAN	36400
80	SA_MAN	61000
110	AC_MGR	12008
90	AD_PRES	24000
60	IT_PROG	28800
100	FI_MGR	12008

Below the table, it says 'More than 10 rows available. Increase rows selector to view more rows.' At the bottom, it says '10 rows returned in 0.00 seconds' and there is a 'Download' link.

8. Display the names and department id of all other employees in the department of an employee.

HomeApplication BuilderSQL WorkshopTeam DevelopmentAdministration

HomeSQL WorkshopSQL Commands

☒ AutocommitRows10SaveRun

```
SELECT
  e1.first_name || ' ' || e1.last_name AS employee_name,
  e1.department_id
FROM employees e1
WHERE e1.department_id = (
  SELECT e2.department_id
  FROM employees e2
  WHERE e2.employee_id = :employee_id
)
AND e1.employee_id != :employee_id;
```

ResultsExplainDescribeSaved SQLHistory

EMPLOYEE_NAME	DEPARTMENT_ID
Steven King	90
Lex De Haan	90

2 rows returned in 0.00 secondsDownload

9. Display name, department id and salary of any employee whose department id and salary of any employee whose department id and salary both match the department id and salary of any employee who earn commission.

Home
Application Builder
SQL Workshop
Team Development
Administration

Home
SQL Workshop
SQL Commands

☒ Autocommit
Rows: 10
Save
Run

```

SELECT
    e1.first_name || ' ' || e1.last_name AS name,
    e1.department_id,
    e1.salary
FROM employees e1
WHERE (e1.department_id, e1.salary) IN (
    SELECT e2.department_id, e2.salary
    FROM employees e2
    WHERE e2.commission_pct IS NOT NULL
);

```



Results
Explain
Describe
Saved SQL
History

NAME	DEPARTMENT_ID	SALARY
John Russell	80	14000
Karen Partners	80	13500
Alberto Errazuriz	80	12000
Ellen Abel	80	11000
Gerald Cambrault	80	11000
Clara Vishney	80	10500
Eleni Zlotkey	80	10500
Harrison Bloom	80	10000
Janette King	80	10000
Peter Tucker	80	10000

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.00 seconds
[Download](#)

11.) Display the details of the employee_id, last name and department id of those employees who live in the cities whose name begins with letter 'T'.

☒ Autocommit Rows 10   Save Run

```
SELECT
  e.employee_id,
  e.last_name,
  d.department_id
FROM employees e
JOIN departments d ON e.department_id = d.department_id
JOIN locations l ON d.location_id = l.location_id
WHERE l.city LIKE 'T%';
```

Results Explain Describe Saved SQL History

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
201	Hartstein	20
202	Fay	20

2 rows returned in 0.01 seconds [Download](#)