• Problem1:

Create a list of all tables whose first two characters in the name of the table is JO −The tables must be owned by the current Oracle User.

Tables Used:

User\_tables

Query:

SELECT table\_name

FROM user\_tables

WHERE table\_name LIKE 'JO%'

ORDER BY table\_name;

• Problem2:

− Create a list that includes the first initial of every employee's first name, a space, and the last name of the employee

• Tables Used: − Employees.

SELECT CONCAT(CONCAT(SUBSTR(first\_name, 1, 1), ' '), last\_name) AS employee\_name

FROM employees;

• Problem3:

− Create a list of every employee's first name concatenated to a space and the employee's last name, and the email of all employees where the email address contains the string 'IN'

• Tables Used: − Employees.

SELECT CONCAT(CONCAT(first\_name, ' '), last\_name) AS full\_name, email

FROM employees

WHERE email LIKE '%IN%';

• Problem4:

− Create a list of 'smallest' last name and the 'highest' last name from the employees table

• Tables Used: − Employees.

SELECT MIN(last\_name) AS smallest\_last\_name, MAX(last\_name) AS highest\_last\_name

FROM employees;

• Problem5:

− Create a list of weekly salaries from the employees table where the weekly salary is between 700 and 3000

− The salaries should be formatted to include a $- sign and have two decimal points like: $9999.99

• Tables Used: − Employees.

SELECT TO\_CHAR(salary, '$9999.99') AS formatted\_salary

FROM employees

WHERE salary BETWEEN 700 AND 3000;

• Problem6:

− Create a list of every employee and his related job title sorted by job\_title

• Tables Used:

− Employees, Jobs.

SELECT SUBSTR(first\_name,1,1)||’ ‘||last\_name AS “Employee Name”,job\_title AS”job

FROM employees e,jobs j

WHERE e.jobs\_id=j.job\_id

ORDER BY job\_title

------or---------

SELECT e.first\_name, e.last\_name, j.job\_title

FROM employees e

JOIN jobs j ON e.job\_id = j.job\_id

ORDER BY j.job\_title;

• Problem7:

−Create a list of every employee’s job, the salary ranges within the job, and the employee's salary −List the lowest and highest salary range within each job with a dash to separate the salaries like this: 100 – 200

• Tables Used: −Employees, Jobs

SELECT SUBSTR(first\_name,1,1)||’ ‘||last\_name AS”Employeename”,job\_titleAS”job”,min\_salary||’\_’||max\_salary AS”salary range”,salary AS”employee’s salary”

FROM employee e,jobs j

WHERE e.job\_id=j.job\_id

ORDER BY job\_title;

------------or----------------------

SELECT

j.job\_title AS job,

TO\_CHAR(MIN(j.min\_salary), '$9999.99') || ' – ' || TO\_CHAR(MAX(j.max\_salary), '$9999.99') AS salary\_range,

TO\_CHAR(e.salary, '$9999.99') AS employee\_salary

FROM

employees e

JOIN

jobs j ON e.job\_id = j.job\_id

GROUP BY

j.job\_title, e.salary

ORDER BY

j.job\_title, e.salary;

Problem8:

− Using an ANSII join method, create a list of every employee's first initial and last name, and department name

− Make sure the tables are joined on all of the foreign keys declared between the two tables

• Tables Used: − Employees, Departments

SELECT SUBSTR(first\_name,1,1)||’ ‘||last\_name AS”EMPLOYEE NAME”,

Department\_name AS”department name”

FROM employee JOIN departments;

----------or--------------

SELECT

SUBSTR(e.first\_name, 1, 1) AS first\_initial,

e.last\_name,

d.department\_name

FROM

employees e

JOIN

departments d ON e.department\_id = d.department\_id;

• Problem9: − Change the previous listing to join only on the department\_id column

• Tables Used: − Employees, Departments

SELECT SUBSTR(first\_name,1,1)||’ ‘||last\_name AS”EMPLOYEE NAME”,

Department\_name AS”department name”

FROM employee JOIN departments

USING (department\_id);

---------or----------------

SELECT

SUBSTR(e.first\_name, 1, 1) AS first\_initial,

e.last\_name,

d.department\_name

FROM

employees e

JOIN

departments d ON e.department\_id = d.department\_id;

• Problem10: − Create a list of every employee's last name, and the word nobody or somebody depending on whether or not the employee has a manager − Use the Oracle DECODE function to create the list

• Tables Used: − Employees

SELECT DECODE(manager\_id,NULL,’Nobody’,’somebody’)AS “works for” last\_name AS”last name”

FROM employee;

------------------or---------------

SELECT

last\_name,

DECODE(manager\_id, NULL, 'nobody', 'somebody') AS manager\_status

FROM

employees;

Problem11:

− Create a list of every employee's first initial and last name, salary, and a yes or no to show whether or not an employee makes a commission.

− Fix this query to produce the result

SELECT SUBSTR(first\_name,1 1)||' '|last\_name, "Employee Name", salary "Salary", DEC(commission\_pct NULL, 'No', 'Yes')'Commission' FROM employees;

• Problem12:

− Create a list of every employee's last name, department name, city, and state\_province − Include departments without employees

− An outer join is required

• Tables Used: − Employees, Departments, Locations.

SELECT

e.last\_name,

d.department\_name,

l.city,

l.state\_province

FROM

departments d

LEFT OUTER JOIN

employees e ON d.department\_id = e.department\_id

LEFT OUTER JOIN

locations l ON d.location\_id = l.location\_id;

• Problem13: −Create a list of every employee's first and last names, and the first occurrence of: commission\_pct, manager\_id, or -1

−If an employee gets commission, display the commission\_pct column; if no commission, then display his manager\_id;

- if he has neither commission nor manager, then the number -1

• Tables Used: −Employees

SELECT

first\_name,

last\_name,

CASE

WHEN commission\_pct IS NOT NULL THEN commission\_pct

WHEN commission\_pct IS NULL AND manager\_id IS NOT NULL THEN manager\_id

ELSE -1

END AS first\_occurrence

FROM

employees;

• Problem14: − Create a list of every employee's last name, salary, and job\_grade for all employees working in departments with a department\_id greater than 50

• Tables Used: − Employees, job\_grades

SELECT

e.last\_name,

e.salary,

jg.job\_grade

FROM

employees e

JOIN

job\_grades jg ON e.job\_grade = jg.job\_grade -- Adjust if the join condition differs

WHERE

e.department\_id > 50;

• Problem15: − Produce a list of every employee's last name and department name

− Include both employees without departments, and departments without employees

• Tables Used: − Employees, Departments

SELECT

e.last\_name,

d.department\_name

FROM

employees e

FULL OUTER JOIN

departments d ON e.department\_id = d.department\_id;

• Problem16:

−Create a treewalking list of every employee's last name, his manager’s last name, and his position in the company

−The top level manager has position 1, this manager's subordinates position 2, their subordinates position 3, and so on −Start the listing with employee number 100

• Tables Used: −Employees

SELECT

e.last\_name AS employee\_last\_name,

m.last\_name AS manager\_last\_name,

LEVEL AS position

FROM

employees e

LEFT JOIN

employees m ON e.manager\_id = m.employee\_id

START WITH

e.employee\_id = 100

CONNECT BY

PRIOR e.employee\_id = e.manager\_id

ORDER BY

position;

• Problem17: − Produce a list of the earliest hire date, the latest hire date, and the number of employees from the employees table

• Tables Used: − Employees

SELECT

MIN(hire\_date) AS earliest\_hire\_date,

MAX(hire\_date) AS latest\_hire\_date,

COUNT(\*) AS number\_of\_employees

FROM

employees;

• Problem18: − Create a list of department names and the departmental costs (salaries added up)

− Include only departments whose salary costs are between 15000 and 31000, and sort the listing by the cost

• Tables Used: − Employees, Departments

SELECT

d.department\_name,

SUM(e.salary) AS departmental\_cost

FROM

employees e

JOIN

departments d ON e.department\_id = d.department\_id

GROUP BY

d.department\_name

HAVING

SUM(e.salary) BETWEEN 15000 AND 31000

ORDER BY

departmental\_cost;

• Problem19: − Create a list of department names, the manager id, manager name (employee last name) of that department, and the average salary in each department

• Tables Used: − Employees, Departments

SELECT

d.department\_name,

e.manager\_id,

m.last\_name AS manager\_name,

AVG(e.salary) AS average\_salary

FROM

departments d

LEFT JOIN

employees e ON d.department\_id = e.department\_id

LEFT JOIN

employees m ON e.manager\_id = m.employee\_id

GROUP BY

d.department\_name, e.manager\_id, m.last\_name;

• Problem20: − Show the highest average salary for the departments in the employees table − Round the result to the nearest whole number

• Tables Used: − Employees

SELECT

ROUND(MAX(average\_salary)) AS highest\_average\_salary

FROM (

SELECT

department\_id,

AVG(salary) AS average\_salary

FROM

employees

GROUP BY

department\_id

) subquery;