SQL QUERIES PRACTICE

V P Jayachitra

Company Database

- Database Schema

EMPLOYEE Table:

Ssn, Fname, Minit, Lname, Bdate, Address, Sex, Salary, Super_ssn, Dno

DEPARTMENT Table:

Dnumber, Dname, Mgr_ssn, Mgr_start_date

PROJECT Table:

Pnumber, Pname, Plocation, Dnum

WORKS_ON Table:

Essn, Pno, Hours

DEPENDENT Table:

Essn, Dependent_name, Sex, Bdate, Relationship

BASIC QUERIES (SELECT, WHERE, ORDER BY)

- Retrieve all employee names and their salaries.
- Find all employees who work in department 5.
- List all employees earning more than $35,000, ordered by salary descending.
- Find all female employees.
- Get all employees whose last name starts with 'W'.
- List all employees born between 1960 and 1970.

EMPLOYEE Table:
Ssn, Fname, Minit, Lname, Bdate, Address, Sex, Salary, Super_ssn, Dno
DEPARTMENT Table:
Dnumber, Dname, Mgr_ssn, Mgr_start_date
PROJECT Table:
Pnumber, Pname, Plocation, Dnum
WORKS_ON Table:
Essn, Pno, Hours
DEPENDENT Table:
Essn, Dependent_name, Sex, Bdate, Relationship

Retrieve all employee names and their salaries.

Solution:

```
SELECT Fname, Lname, Salary FROM EMPLOYEE;
```

Find all employees who work in department 5.

Solution:

```
SELECT Fname, Lname, Dno FROM EMPLOYEE WHERE Dno = 5;
```

List all employees earning more than $35,000, ordered by salary descending.

Solution:

```
SELECT Fname, Lname, Salary FROM EMPLOYEE WHERE Salary > 35000 ORDER BY Salary DESC;
```

Find all female employees.

Solution:

```
SELECT Fname, Lname, Sex FROM EMPLOYEE WHERE Sex = 'F';
```

Get all employees whose last name starts with 'W'.

Solution:

```
SELECT Fname, Lname FROM EMPLOYEE WHERE Lname LIKE 'W%';
```

List all employees born between 1960 and 1970.

Solution:

```
SELECT Fname, Lname, Bdate FROM EMPLOYEE WHERE Bdate BETWEEN '1960-01-01' AND '1970-12-31';
```

EMPLOYEE Table:
Ssn, Fname, Minit, Lname, Bdate, Address, Sex, Salary, Super_ssn, Dno
DEPARTMENT Table:
Dnumber, Dname, Mgr_ssn, Mgr_start_date
PROJECT Table:
Pnumber, Pname, Plocation, Dnum
WORKS_ON Table:
Essn, Pno, Hours
DEPENDENT Table:
Essn, Dependent_name, Sex, Bdate, Relationship

JOINS

- List all employees with their department names.
- Find each employee's name along with their supervisor's name.
- List all projects with their controlling department names.
- Show each employee's name, project name, and hours worked.
- Find all employees with their dependents' names (include employees with no dependents).

EMPLOYEE Table:
Ssn, Fname, Minit, Lname, Bdate, Address, Sex, Salary, Super_ssn, Dno
DEPARTMENT Table:
Dnumber, Dname, Mgr_ssn, Mgr_start_date
PROJECT Table:
Pnumber, Pname, Plocation, Dnum
WORKS_ON Table:
Essn, Pno, Hours
DEPENDENT Table:
Essn, Dependent_name, Sex, Bdate, Relationship

EMPLOYEE Table:
Ssn, Fname, Minit, Lname, Bdate, Address, Sex, Salary, Super_ssn, Dno
DEPARTMENT Table:
Dnumber, Dname, Mgr_ssn, Mgr_start_date
PROJECT Table:
Pnumber, Pname, Plocation, Dnum
WORKS_ON Table:
Essn, Pno, Hours
DEPENDENT Table:
Essn, Dependent_name, Sex, Bdate, Relationship

List all employees with their department names.

Solution:

```
SELECT E.Fname, E.Lname, D.Dname FROM EMPLOYEE E

JOIN DEPARTMENT D ON E.Dno = D.Dnumber;
```

Find each employee's name along with their supervisor's name.

Solution:

```
SELECT E.Fname AS Employee_First, E.Lname AS Employee_Last,        S.Fname AS Supervisor_First, S.Lname AS
Supervisor_Last FROM EMPLOYEE E LEFT JOIN EMPLOYEE S ON E.Super_ssn = S.Ssn;
```

List all projects with their controlling department names.

Solution:

```
SELECT P.Pname, P.Plocation, D.Dname FROM PROJECT P JOIN DEPARTMENT D ON P.Dnum = D.Dnumber;
```

Show each employee's name, project name, and hours worked.

Solution:

```
SELECT E.Fname, E.Lname, P.Pname, W.Hours FROM EMPLOYEE E JOIN WORKS_ON W ON E.Ssn = W.Essn JOIN PROJECT P
ON W.Pno = P.Pnumber;
```

Find all employees with their dependents' names (include employees with no dependents).

Solution:

```
SELECT E.Fname, E.Lname, D.Dependent_name, D.Relationship FROM EMPLOYEE E LEFT JOIN DEPENDENT D ON E.Ssn =
D.Essn;
```

AGGREGATE FUNCTIONS (COUNT, SUM, AVG, MAX, MIN)

1. Count the total number of employees.
2. Find the average salary of all employees.
3. What is the highest salary in the company?
4. Calculate the total hours worked on project number 1.
5. Find the minimum salary in department 5.

EMPLOYEE Table:
Ssn, Fname, Minit, Lname, Bdate, Address, Sex, Salary, Super_ssn, Dno
DEPARTMENT Table:
Dnumber, Dname, Mgr_ssn, Mgr_start_date
PROJECT Table:
Pnumber, Pname, Plocation, Dnum
WORKS_ON Table:
Essn, Pno, Hours
DEPENDENT Table:
Essn, Dependent_name, Sex, Bdate, Relationship

Count the total number of employees.

Solution:

```
SELECT COUNT(*) AS Total_Employees FROM EMPLOYEE;
```

Find the average salary of all employees.

Solution:

```
SELECT AVG(Salary) AS Average_Salary FROM EMPLOYEE;
```

What is the highest salary in the company?

Solution:

```
SELECT MAX(Salary) AS Highest_Salary FROM EMPLOYEE;
```

Calculate the total hours worked on project number 1.

Solution:

```
SELECT SUM(Hours) AS Total_Hours FROM WORKS_ON WHERE Pno = 1;
```

Find the minimum salary in department 5.

Solution:

```
SELECT MIN(Salary) AS Minimum_Salary FROM EMPLOYEE WHERE Dno = 5;
```

EMPLOYEE Table:
Ssn, Fname, Minit, Lname, Bdate, Address, Sex, Salary, Super_ssn, Dno
DEPARTMENT Table:
Dnumber, Dname, Mgr_ssn, Mgr_start_date
PROJECT Table:
Pnumber, Pname, Plocation, Dnum
WORKS_ON Table:
Essn, Pno, Hours
DEPENDENT Table:
Essn, Dependent_name, Sex, Bdate, Relationship

GROUP BY & HAVING

- Count the number of employees in each department.
- Find the average salary for each department.
- List departments with more than 2 employees.
- For each project, show the project name and total hours worked on it.
- Find employees who have more than 2 dependents.
- Show departments where the average salary is greater than $35,000.

EMPLOYEE Table:
Ssn, Fname, Minit, Lname, Bdate, Address, Sex, Salary, Super_ssn, Dno
DEPARTMENT Table:
Dnumber, Dname, Mgr_ssn, Mgr_start_date
PROJECT Table:
Pnumber, Pname, Plocation, Dnum
WORKS_ON Table:
Essn, Pno, Hours
DEPENDENT Table:
Essn, Dependent_name, Sex, Bdate, Relationship

EMPLOYEE Table:
Ssn, Fname, Minit, Lname, Bdate, Address, Sex, Salary, Super_ssn, Dno
DEPARTMENT Table:
Dnumber, Dname, Mgr_ssn, Mgr_start_date
PROJECT Table:
Pnumber, Pname, Plocation, Dnum
WORKS_ON Table:
Essn, Pno, Hours
DEPENDENT Table:
Essn, Dependent_name, Sex, Bdate, Relationship

1. **Count the number of employees in each department.**

```
SELECT Dno,COUNT(*) AS EmpCount FROM EMPLOYEE GROUP BY Dno;
```

1. **Find the average salary for each department.**

Solution:

```
SELECT D.Dname, AVG(E.Salary) AS Avg_Salary FROM EMPLOYEE E JOIN DEPARTMENT D ON E.Dno = D.Dnumber GROUP BY
D.Dname;
```

1. **List departments with more than 2 employees.**

Solution:

```
SELECT Dno, COUNT(*) AS Employee_Count FROM EMPLOYEE GROUP BY Dno HAVING COUNT(*) > 2;
```

1. **For each project, show the project name and total hours worked on it.**

Solution:
```
SELECT P.Pname, SUM(W.Hours) AS Total_Hours FROM PROJECT P
JOIN WORKS_ON W ON P.Pnumber = W.Pno GROUP BY P.Pname;
```

1. **Find employees who have more than 2 dependents.**

Solution:

```
SELECT E.Fname, E.Lname, COUNT(D.Dependent_name) AS Num_Dependents FROM EMPLOYEE E JOIN DEPENDENT D ON E.Ssn =
D.Essn GROUP BY E.Ssn, E.Fname, E.Lname HAVING COUNT(D.Dependent_name) > 2;
```

1. **Show departments where the average salary is greater than $35,000.**

Solution:

```
SELECT D.Dname, AVG(E.Salary) AS Avg_Salary FROM EMPLOYEE E JOIN DEPARTMENT D ON E.Dno = D.Dnumber GROUP BY
D.Dname HAVING AVG(E.Salary) > 35000;
```

SUBQUERIES - SINGLE ROW

- Find employees who earn more than the average salary.
- Find the employee with the highest salary.
- List employees who work in the same department as 'John Smith'.
- Find employees earning less than Jennifer Wallace.

EMPLOYEE Table:
Ssn, Fname, Minit, Lname, Bdate, Address, Sex, Salary, Super_ssn, Dno
DEPARTMENT Table:
Dnumber, Dname, Mgr_ssn, Mgr_start_date
PROJECT Table:
Pnumber, Pname, Plocation, Dnum
WORKS_ON Table:
Essn, Pno, Hours
DEPENDENT Table:
Essn, Dependent_name, Sex, Bdate, Relationship

1. **Find employees who earn more than the average salary.**

```
SELECT Fname, Lname, Salary FROM EMPLOYEE WHERE Salary > (SELECT AVG(Salary) FROM EMPLOYEE);
```

1. **Find the employee with the highest salary.**

Solution:

```
SELECT Fname, Lname, Salary FROM EMPLOYEE WHERE Salary = (SELECT MAX(Salary) FROM EMPLOYEE);
```

1. **List employees who work in the same department as 'John Smith'.**

Solution:

```
SELECT Fname, Lname, Dno FROM EMPLOYEE WHERE Dno = (SELECT Dno FROM EMPLOYEE
WHERE Fname = 'John' AND Lname = 'Smith');
```

1. **Find employees earning less than Jennifer Wallace.**

Solution:

```
SELECT Fname, Lname, Salary FROM EMPLOYEE WHERE Salary < (SELECT Salary FROM EMPLOYEE
WHERE Fname = 'Jennifer' AND Lname = 'Wallace');
```

EMPLOYEE Table:
Ssn, Fname, Minit, Lname, Bdate, Address, Sex, Salary, Super_ssn, Dno
DEPARTMENT Table:
Dnumber, Dname, Mgr_ssn, Mgr_start_date
PROJECT Table:
Pnumber, Pname, Plocation, Dnum
WORKS_ON Table:
Essn, Pno, Hours
DEPENDENT Table:
Essn, Dependent_name, Sex, Bdate, Relationship

SUBQUERIES - MULTIPLE ROW (IN, ANY, ALL)

- Find employees who work on project 'ProductX'.
- List employees who are not working on any project.
- Find employees whose salary is greater than the salary of ANY employee in department 4.
- Find employees whose salary is greater than ALL employees in department 4.
- Find department names that have projects.

EMPLOYEE Table:
Ssn, Fname, Minit, Lname, Bdate, Address, Sex, Salary, Super_ssn, Dno
DEPARTMENT Table:
Dnumber, Dname, Mgr_ssn, Mgr_start_date
PROJECT Table:
Pnumber, Pname, Plocation, Dnum
WORKS_ON Table:
Essn, Pno, Hours
DEPENDENT Table:
Essn, Dependent_name, Sex, Bdate, Relationship

Find employees who work on project 'ProductX'.

Solution:

```
SELECT Fname, Lname FROM EMPLOYEE WHERE Ssn IN (SELECT Essn FROM WORKS_ON WHERE Pno =

(SELECT Pnumber FROM PROJECT WHERE Pname = 'ProductX'));
```

List employees who are not working on any project.

Solution:

```
SELECT Fname, Lname FROM EMPLOYEE WHERE Ssn NOT IN (SELECT DISTINCT Essn FROM WORKS_ON);
```

Find employees whose salary is greater than the salary of ANY employee in department 4.

Solution:

```
SELECT Fname, Lname, Salary FROM EMPLOYEE WHERE Salary > ANY (SELECT Salary FROM EMPLOYEE WHERE Dno = 4);
```

Find employees whose salary is greater than ALL employees in department 4.

Solution:

```
SELECT Fname, Lname, Salary

FROM EMPLOYEE WHERE Salary > ALL

(SELECT Salary FROM EMPLOYEE WHERE Dno = 4);
```

EMPLOYEE Table:
Ssn, Fname, Minit, Lname, Bdate, Address, Sex, Salary, Super_ssn, Dno
DEPARTMENT Table:
Dnumber, Dname, Mgr_ssn, Mgr_start_date
PROJECT Table:
Pnumber, Pname, Plocation, Dnum
WORKS_ON Table:
Essn, Pno, Hours
DEPENDENT Table:
Essn, Dependent_name, Sex, Bdate, Relationship

Find department names that have projects.

Solution:

```
SELECT Dname FROM DEPARTMENT WHERE Dnumber IN (SELECT DISTINCT Dnum FROM PROJECT);
```

CORRELATED SUBQUERIES (EXISTS & NOT EXISTS)

- Find employees who have at least one dependent.
- Find employees who have NO dependents.
- Find projects that have at least one employee working more than 30 hours on it.
- Find departments that have no employees.
- Find employees who work on ALL projects controlled by their department.

EMPLOYEE Table:
Ssn, Fname, Minit, Lname, Bdate, Address, Sex, Salary, Super_ssn, Dno
DEPARTMENT Table:
Dnumber, Dname, Mgr_ssn, Mgr_start_date
PROJECT Table:
Pnumber, Pname, Plocation, Dnum
WORKS_ON Table:
Essn, Pno, Hours
DEPENDENT Table:
Essn, Dependent_name, Sex, Bdate, Relationship

1. **Find employees who have at least one dependent.**

Solution:

```
SELECT Fname, Lname FROM EMPLOYEE E WHERE EXISTS (SELECT * FROM DEPENDENT D WHERE D.Essn = E.Ssn);
```

1. **Find employees who have NO dependents.**

Solution:

```
SELECT Fname, Lname FROM EMPLOYEE E WHERE NOT EXISTS (SELECT * FROM DEPENDENT D WHERE D.Essn = E.Ssn);
```

1. **Find projects that have at least one employee working more than 30 hours on it.**

Solution:

```
SELECT Pname FROM PROJECT P WHERE EXISTS(SELECT * FROM WORKS_ON W WHERE W.Pno= P.Pnumber AND W.Hours > 30);
```

1. **Find departments that have no employees.**

Solution:

```
SELECT Dname FROM DEPARTMENT D WHERE NOT EXISTS

(SELECT * FROM EMPLOYEE E WHERE E.Dno = D.Dnumber);
```

1. **Find employees who work on ALL projects controlled by their department.**

Solution:

```
SELECT E.Fname, E.Lname FROM EMPLOYEE E WHERE NOT EXISTS (SELECT * FROM PROJECT P WHERE P.Dnum = E.Dno
AND NOT EXISTS (SELECT * FROM WORKS_ON W WHERE W.Essn = E.Ssn AND W.Pno = P.Pnumber));
```

EMPLOYEE Table:
Ssn, Fname, Minit, Lname, Bdate, Address, Sex, Salary, Super_ssn, Dno
DEPARTMENT Table:
Dnumber, Dname, Mgr_ssn, Mgr_start_date
PROJECT Table:
Pnumber, Pname, Plocation, Dnum
WORKS_ON Table:
Essn, Pno, Hours
DEPENDENT Table:
Essn, Dependent_name, Sex, Bdate, Relationship

ADVANCED QUERIES

- Find employees who work on at least 2 projects.
- List all managers (employees who manage departments).
- Find the department with the highest total salary expenditure.
- Find employees who earn more than their supervisor.

EMPLOYEE Table:
Ssn, Fname, Minit, Lname, Bdate, Address, Sex, Salary, Super_ssn, Dno
DEPARTMENT Table:
Dnumber, Dname, Mgr_ssn, Mgr_start_date
PROJECT Table:
Pnumber, Pname, Plocation, Dnum
WORKS_ON Table:
Essn, Pno, Hours
DEPENDENT Table:
Essn, Dependent_name, Sex, Bdate, Relationship

Find employees who work on at least 2 projects.

Solution:

```
SELECT E.Fname, E.Lname, COUNT(W.Pno) AS Num_Projects

FROM EMPLOYEE E JOIN WORKS_ON W ON E.Ssn = W.Essn GROUP BY E.Ssn, E.Fname, E.Lname HAVING COUNT(W.Pno) >= 2;
```

List all managers (employees who manage departments).

Solution:

```
SELECT E.Fname, E.Lname, D.Dname FROM EMPLOYEE E JOIN DEPARTMENT D ON E.Ssn = D.Mgr_ssn;
```

Find the department with the highest total salary expenditure.

Solution:

```
SELECT D.Dname, SUM(E.Salary) AS Total_Salary FROM EMPLOYEE E JOIN DEPARTMENT D ON E.Dno = D.Dnumber GROUP BY
D.Dname ORDER BY Total_Salary DESC FETCH FIRST 1 ROW ONLY;
```

Find employees who earn more than their supervisor.

Solution:

```
SELECT E.Fname, E.Lname, E.Salary AS Emp_Salary,

S.Fname AS Sup_Fname, S.Lname AS Sup_Lname, S.Salary AS Sup_Salary

FROM EMPLOYEE E JOIN EMPLOYEE S ON E.Super_ssn = S.Ssn WHERE E.Salary > S.Salary;
```

EMPLOYEE Table:
Ssn, Fname, Minit, Lname, Bdate, Address, Sex, Salary, Super_ssn, Dno
DEPARTMENT Table:
Dnumber, Dname, Mgr_ssn, Mgr_start_date
PROJECT Table:
Pnumber, Pname, Plocation, Dnum
WORKS_ON Table:
Essn, Pno, Hours
DEPENDENT Table:
Essn, Dependent_name, Sex, Bdate, Relationship

ADVANCED QUERIES contd..

- List projects located in Houston with their controlling department.
- Find employees who have a dependent with the same first name as them.
- Calculate the total hours each employee works across all projects.
- For each employee, show their name, salary, and how much they earn compared to the average.
- Show each department's average salary compared to the company average.

EMPLOYEE Table:
Ssn, Fname, Minit, Lname, Bdate, Address, Sex, Salary, Super_ssn, Dno
DEPARTMENT Table:
Dnumber, Dname, Mgr_ssn, Mgr_start_date
PROJECT Table:
Pnumber, Pname, Plocation, Dnum
WORKS_ON Table:
Essn, Pno, Hours
DEPENDENT Table:
Essn, Dependent_name, Sex, Bdate, Relationship

List projects located in Houston with their controlling department.

Solution:

```
SELECT P.Pname, P.Plocation, D.Dname FROM PROJECT P JOIN
DEPARTMENT D ON P.Dnum = D.Dnumber WHERE P.Plocation =
'Houston';
```

Find employees who have a dependent with the same first name as them.

Solution:

```
SELECT DISTINCT E.Fname, E.Lname FROM EMPLOYEE E JOIN DEPENDENT
D ON E.Ssn = D.Essn WHERE E.Fname = D.Dependent_name;
```

Calculate the total hours each employee works across all projects.

Solution:

```
SELECT E.Fname, E.Lname, SUM(W.Hours) AS Total_Hours FROM
EMPLOYEE E JOIN WORKS_ON W ON E.Ssn = W.Essn GROUP BY E.Ssn,
E.Fname, E.Lname ORDER BY Total_Hours DESC;
```

EMPLOYEE Table:
Ssn, Fname, Minit, Lname, Bdate, Address, Sex, Salary, Super_ssn, Dno

DEPARTMENT Table:
Dnumber, Dname, Mgr_ssn, Mgr_start_date

PROJECT Table:
Pnumber, Pname, Plocation, Dnum

WORKS_ON Table:
Essn, Pno, Hours

DEPENDENT Table:
Essn, Dependent_name, Sex, Bdate, Relationship

For each employee, show their name, salary, and how much they earn compared to the average.

Solution:

```
SELECT Fname, Lname, Salary, Salary
- (SELECT AVG(Salary) FROM EMPLOYEE) AS
Diff_From_Avg FROM EMPLOYEE;
```

Show each department's average salary compared to the company average.

Solution:

```
SELECT D.Dname, AVG(E.Salary) AS
Dept_Avg, (SELECT AVG(Salary) FROM
EMPLOYEE) AS Company_Avg,
AVG(E.Salary) - (SELECT AVG(Salary) FROM
EMPLOYEE) AS Difference FROM EMPLOYEE E
JOIN DEPARTMENT D ON E.Dno = D.Dnumber
GROUP BY D.Dname;
```

EMPLOYEE Table:
Ssn, Fname, Minit, Lname, Bdate, Address, Sex, Salary, Super_ssn, Dno
DEPARTMENT Table:
Dnumber, Dname, Mgr_ssn, Mgr_start_date
PROJECT Table:
Pnumber, Pname, Plocation, Dnum
WORKS_ON Table:
Essn, Pno, Hours
DEPENDENT Table:
Essn, Dependent_name, Sex, Bdate, Relationship

Note

FETCH FIRST n ROWS ONLY	Get first n rows
FETCH FIRST n ROWS WITH TIES	Include ties at nth position
FETCH FIRST n PERCENT ROWS ONLY	Get top n% of rows
OFFSET m ROWS FETCH NEXT n ROWS ONLY	Skip m, get next n (pagination)

Get the first n rows:

```
SELECT Fname, Lname, Salary
FROM EMPLOYEE
ORDER BY Salary DESC
FETCH FIRST 5 ROWS ONLY;
```

```
SELECT columns
FROM table
ORDER BY column
FETCH FIRST n ROWS ONLY;
```

Get first n rows, plus any ties:

If 2rd and 3th employees have the same salary, **both are included**.

```
SELECT Fname, Lname, Salary
FROM EMPLOYEE
ORDER BY Salary DESC
FETCH FIRST 3 ROWS WITH TIES;
```

```
SELECT columns
FROM table
ORDER BY column
OFFSET m ROWS FETCH NEXT n ROWS ONLY;
```

Skip m rows, then fetch n rows:

```
SELECT Fname, Lname, Salary
FROM EMPLOYEE
ORDER BY Salary DESC
OFFSET 5 ROWS FETCH NEXT 5 ROWS ONLY;
```

View

- A **View** is a virtual table.
- It doesn't store data itself
- It’s essentially a **stored query** that the database engine executes every time you reference the view name.
- Views provide data abstraction, security, and simplify complex queries.

Create the view

```
CREATE VIEW View_Name AS
SELECT column1, column2…
FROM Table_Name
WHERE condition;
```

Query the view

```
    SELECT * FROM view_name;
```

Drop the view

```
    DROP VIEW view_name;
```

View

Benefits of Views:

- Security: Hide sensitive columns from users
- Simplification: Complex JOINs can be encapsulated
- Logical data independence: Base table changes don't affect applications

Updatable vs Non-Updatable Views:

- Updatable: Single table, no aggregates, no DISTINCT, no GROUP BY
- Non-updatable: Multiple tables (JOINs), aggregates, DISTINCT, GROUP BY

Updatable View

- INSERT / UPDATE / DELETE (The "Updatable View" Rule)
- To update a view, the RDBMS must know exactly which one row in the physical table to change.
  - When it works: The view is defined on a single table and includes the Primary Key.
  - When it fails: The view contains JOINS, DISTINCT, GROUP BY, or AGGREGATES.

When Rows are Affected

- Without CHECK OPTION: The row is added to the base table but "disappears" from your view.
- With CHECK OPTION: The database rejects the insert to maintain integrity.

```
CREATE [OR REPLACE] VIEW view_name [(alias1, alias2...)] AS
SELECT column_list
FROM table_list
[WHERE condition]
[WITH CHECK OPTION [CONSTRAINT constraint_name]]
[WITH READ ONLY];
```

```
SQL> desc emp2
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPNO                                     NOT NULL NUMBER(4)
 SALARY                                             NUMBER(8,2)
 SQL> select * from emp2
     EMPNO     SALARY
---------- ----------
       100      20000
       101      40000
 SQL> create or replace view emp_view(no, sal) as select * from emp2;
 View created.
 SQL> desc emp_view;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 NO                                        NOT NULL NUMBER(4)
 SAL                                                NUMBER(8,2)
```

```
SQL> insert into emp_view values(102, 30000);

1 row created.

SQL> select * from emp_view;

        NO        SAL
---------- ----------
       100      20000
       101      40000
       102      30000

SQL> select * from emp2;

     EMPNO     SALARY
---------- ----------
       100      20000
       101      40000
       102      30000
```

```
SQL> insert into emp2 values(103, 50000);

1 row created.

SQL> select * from emp2;

     EMPNO     SALARY
---------- ----------
       100      20000
       101      40000
       102      30000
       103      50000

SQL> select * from emp_view;

        NO        SAL
---------- ----------
       100      20000
       101      40000
       102      30000
       103      50000
)
```

```
SQL> create or replace view empv as select * from
emp2 where salary >40000;

View created.

SQL> select * from empv;

     EMPNO     SALARY
---------- ----------
       103      50000

SQL> insert into empv values(105, 20000);

1 row created.
SQL> select * from empv;

     EMPNO     SALARY
---------- ----------
       103      50000
```

```
SQL> select * from emp2;

     EMPNO     SALARY
---------- ----------
       100      20000
       101      40000
       102      30000
       103      50000
       105      20000

SQL> create or replace view empv as select * from emp2
where salary >40000 with check option;

View created.

SQL> insert into empv values(105, 20000);
insert into empv values(105, 20000)
            *
ERROR at line 1:
ORA-01402: view WITH CHECK OPTION where-clause
violation
```

PL/SQL

```
SQL> set serveroutput on
SQL> declare
  2  sal emp2.salary%type;
  3  begin
  4  select salary into sal from emp2 where empno=102;
  5  dbms_output.put_line('sal:' || sal);
  6  end;
  7  /
sal:30000

PL/SQL procedure successfully completed.
```