```
        DBMS_OUTPUT.PUT_LINE('Employee ID: ' || emp_record.employee_id);
        DBMS_OUTPUT.PUT_LINE('Employee Name: ' || emp_record.employee_name);
        DBMS_OUTPUT.PUT_LINE('Department Name: ' || NVL(emp_record.department_name, 'No Department'));
        DBMS_OUTPUT.PUT_LINE('-------------------------');

        FETCH emp_cursor INTO emp_record;
    END LOOP;

    CLOSE emp_cursor;
END;
/

-- PROGRAM 13: Display job IDs, titles, and minimum salaries of all jobs.
DECLARE
    CURSOR job_cursor IS
        SELECT job_id, job_title, min_salary
        FROM jobs;  -- Assuming a jobs table exists

    job_record job_cursor%ROWTYPE;
BEGIN
    OPEN job_cursor;
    FETCH job_cursor INTO job_record;

    WHILE job_cursor%FOUND LOOP
        DBMS_OUTPUT.PUT_LINE('Job ID: ' || job_record.job_id);
        DBMS_OUTPUT.PUT_LINE('Job Title: ' || job_record.job_title);
        DBMS_OUTPUT.PUT_LINE('Minimum Salary: ' || job_record.min_salary);
        DBMS_OUTPUT.PUT_LINE('-------------------------');

        FETCH job_cursor INTO job_record;
    END LOOP;

    CLOSE job_cursor;
END;
/
```

STDIN

Input for the program (Optional)

```
Value of myvariable2: 200
Salary adjusted for employee with ID 122.
Employee Name: Alice
Small Number: 10
Large Number: 20
Number of employees in department 50: 2
There are vacancies in department 50.
Number of employees in department 50: 2
Vacancies available: 43
Employee ID | Full Name                | Job Title | Hire Date | Salary
-------------------------------------------------------------------------
110 | John Doe                  |   IT_PROG | 10-JAN-22 |  60000
122 | Jane Smith                |    HR_REP | 15-MAY-21 |  55000
150 | Alice Johnson             |    HR_REP | 20-MAR-20 |  45000
Employee ID: 110
Employee Name: John Doe
Department Name: Human Resources
-------------------------
Employee ID: 122
Employee Name: Jane Smith
Department Name: Human Resources
-------------------------
Employee ID: 150
```

Input

Run SQL

```sql
-- For SQLite or similar systems
CREATE TABLE departments (
    department_id INTEGER PRIMARY KEY,
    department_name TEXT
);

-- Insert example data
INSERT INTO departments (department_id, department_name) VALUES (500, 'Education');
INSERT INTO departments (department_id, department_name) VALUES (510, 'Human Resources');

-- Query all rows
SELECT * FROM departments;

-- Remove the rows
DELETE FROM departments WHERE department_id = 500;
```

Output

Available Tables

Customers

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

Departments

department_id	department_name
empty	

```sql
-- Insert example data
INSERT INTO departments (department_id, department_name) VALUES (500, 'Education');
INSERT INTO departments (department_id, department_name) VALUES (510, 'Human Resources');

-- Query all rows
SELECT * FROM departments;

-- Remove the rows
DELETE FROM departments WHERE department_id = 500;
DELETE FROM departments WHERE department_id = 510;

-- Commit is not needed in SQLite but can be used if in a transaction
```

Output

Error: table departments already exists