“Año de la recuperación y consolidación de la economía peruana”

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

Facultad de Ingeniería de Sistemas e Informática

Escuela profesional de Ingeniería de Software

Laboratorio 06

Asignatura: Base de Datos II

Docente: Jorge Luis Chávez Soto

Alumno:

- Chavez Gave, Jose Luis

Fecha: 27 de octubre

Lima - Perú

2025

1. Creación de Tablespaces y carga de datos

```
-- Tabla: REGIONS
CREATE TABLE regions (
  region_id     NUMBER PRIMARY KEY,
  region_name   VARCHAR2(25)
);

-- Tabla: COUNTRIES
CREATE TABLE countries (
  country_id    CHAR(2) PRIMARY KEY,
  country_name  VARCHAR2(40),
  region_id     NUMBER,
  CONSTRAINT fk_region FOREIGN KEY (region_id) REFERENCES regions(region_id)
);

-- Tabla: LOCATIONS
CREATE TABLE locations (
  location_id     NUMBER(4) PRIMARY KEY,
  street_address  VARCHAR2(40),
  postal_code     VARCHAR2(12),
  city            VARCHAR2(30) NOT NULL,
  state_province  VARCHAR2(25),
  country_id      CHAR(2),
  CONSTRAINT fk_country FOREIGN KEY (country_id) REFERENCES
countries(country_id)
);
```

```
-- Tabla: DEPARTMENTS
CREATE TABLE departments (
  department_id    NUMBER(4) PRIMARY KEY,
  department_name  VARCHAR2(30) NOT NULL,
  manager_id       NUMBER(6),
  location_id      NUMBER(4),
  CONSTRAINT fk_location FOREIGN KEY (location_id) REFERENCES
locations(location_id)
);

-- Tabla: JOBS
CREATE TABLE jobs (
  job_id      VARCHAR2(10) PRIMARY KEY,
  job_title   VARCHAR2(35) NOT NULL,
  min_salary  NUMBER(6),
  max_salary  NUMBER(6)
);

-- Tabla: EMPLOYEES
CREATE TABLE employees (
  employee_id     NUMBER(6) PRIMARY KEY,
  first_name      VARCHAR2(20),
  last_name       VARCHAR2(25) NOT NULL,
  email           VARCHAR2(25) NOT NULL UNIQUE,
  phone_number    VARCHAR2(20),
```

```
  hire_date       DATE NOT NULL,
  job_id          VARCHAR2(10) NOT NULL,
  salary          NUMBER(8,2),
  commission_pct  NUMBER(2,2),
  manager_id      NUMBER(6),
  department_id   NUMBER(4),
  CONSTRAINT fk_job FOREIGN KEY (job_id) REFERENCES jobs(job_id),
  CONSTRAINT fk_dept FOREIGN KEY (department_id) REFERENCES
departments(department_id)
);

-- Tabla: JOB_HISTORY
CREATE TABLE job_history (
  employee_id    NUMBER(6) NOT NULL,
  start_date     DATE NOT NULL,
  end_date       DATE NOT NULL,
  job_id         VARCHAR2(10) NOT NULL,
  department_id  NUMBER(4),
  CONSTRAINT pk_job_history PRIMARY KEY (employee_id, start_date),
  CONSTRAINT fk_jh_employee FOREIGN KEY (employee_id) REFERENCES
employees(employee_id),
  CONSTRAINT fk_jh_job FOREIGN KEY (job_id) REFERENCES jobs(job_id),
  CONSTRAINT fk_jh_dept FOREIGN KEY (department_id) REFERENCES
departments(department_id)
);
```

```
-- Tabla: HORARIO
CREATE TABLE horario (
  dia_semana     VARCHAR2(10),
  turno          VARCHAR2(10),
  hora_inicio    DATE,
  hora_termino   DATE
);

-- Tabla: EMPLEADO_HORARIO
CREATE TABLE empleado_horario (
  dia_semana     VARCHAR2(10),
  turno          VARCHAR2(10),
  empleado_id    NUMBER(6),
  CONSTRAINT fk_eh_empleado FOREIGN KEY (empleado_id) REFERENCES
employees(employee_id)
);

-- Tabla: ASISTENCIA_EMPLEADO
CREATE TABLE asistencia_empleado (
  empleado_id        NUMBER(6),
  dia_semana         VARCHAR2(10),
  fecha              DATE,
  hora_inicio_real   DATE,
  hora_termino_real  DATE,
  CONSTRAINT fk_ae_empleado FOREIGN KEY (empleado_id) REFERENCES
employees(employee_id)
```

```
);

-- Tabla: CAPACITACION
CREATE TABLE capacitacion (
  cod_capacitacion   NUMBER PRIMARY KEY,
  nombre             VARCHAR2(50),
  horas              NUMBER,
  descripcion        VARCHAR2(200)
);

-- Tabla: EMPLEADO_CAPACITACION
CREATE TABLE empleado_capacitacion (
  empleado_id        NUMBER(6),
  cod_capacitacion   NUMBER,
  CONSTRAINT fk_ec_empleado FOREIGN KEY (empleado_id) REFERENCES
employees(employee_id),
  CONSTRAINT fk_ec_capacitacion FOREIGN KEY (cod_capacitacion) REFERENCES
capacitacion(cod_capacitacion)
);
```

2. Inserción de datos

2.1. Regions

```
INSERT INTO regions VALUES (1, 'Europe');
INSERT INTO regions VALUES (2, 'Americas');
INSERT INTO regions VALUES (3, 'Asia');
INSERT INTO regions VALUES (4, 'Africa');
```

2.2. Countries

```
INSERT INTO countries VALUES ('FR', 'France', 1);
INSERT INTO countries VALUES ('US', 'United States', 2);
INSERT INTO countries VALUES ('JP', 'Japan', 3);
INSERT INTO countries VALUES ('BR', 'Brazil', 2);
INSERT INTO countries VALUES ('EG', 'Egypt', 4);
```

2.3. Locations

```
INSERT INTO locations VALUES (1000, 'Champs Elysees', '75008', 'Paris',
'Ile-de-France', 'FR');
INSERT INTO locations VALUES (1001, '5th Avenue', '10001', 'New York', 'NY',
'US');
INSERT INTO locations VALUES (1002, 'Shibuya', '150-0002', 'Tokyo', 'Tokyo',
'JP');
INSERT INTO locations VALUES (1003, 'Paulista', '01310-000', 'São Paulo', 'SP',
'BR');
INSERT INTO locations VALUES (1004, 'Zamalek', '11211', 'Cairo', 'Cairo', 'EG');
```

2.4. Departments

```
INSERT INTO departments VALUES (10, 'Sales', NULL, 1001);
INSERT INTO departments VALUES (20, 'Marketing', NULL, 1002);
INSERT INTO departments VALUES (30, 'IT', NULL, 1000);
INSERT INTO departments VALUES (40, 'HR', NULL, 1003);
INSERT INTO departments VALUES (50, 'Finance', NULL, 1004);
```

2.5. Jobs

```
INSERT INTO jobs VALUES ('SA_REP', 'Sales Representative', 3000, 7000);
INSERT INTO jobs VALUES ('MK_MAN', 'Marketing Manager', 4000, 9000);
INSERT INTO jobs VALUES ('IT_PROG', 'Programmer', 3500, 8000);
INSERT INTO jobs VALUES ('HR_ASS', 'HR Assistant', 2500, 6000);
INSERT INTO jobs VALUES ('FI_ANAL', 'Financial Analyst', 4500, 9500);
```

2.6. Employees

```
INSERT INTO employees VALUES (101, 'Alice', 'Smith', 'asmith@example.com',
'555-1234', TO_DATE('2018-01-15','YYYY-MM-DD'), 'SA_REP', 5000, NULL,
NULL, 10);
INSERT INTO employees VALUES (102, 'Bob', 'Jones', 'bjones@example.com',
'555-2345', TO_DATE('2019-03-10','YYYY-MM-DD'), 'MK_MAN', 6000, NULL,
NULL, 20);
INSERT INTO employees VALUES (103, 'Carlos', 'Tanaka', 'ctanaka@example.com',
'555-3456', TO_DATE('2020-06-01','YYYY-MM-DD'), 'IT_PROG', 5500, NULL,
NULL, 30);
INSERT INTO employees VALUES (104, 'Diana', 'Silva', 'dsilva@example.com',
'555-4567', TO_DATE('2021-09-20','YYYY-MM-DD'), 'HR_ASS', 3000, NULL,
NULL, 40);
INSERT INTO employees VALUES (105, 'Ethan', 'Khan', 'ekhan@example.com',
'555-5678', TO_DATE('2022-11-05','YYYY-MM-DD'), 'FI_ANAL', 7000, NULL,
NULL, 50);
-- 5 adicionales para pruebas de rotación y asistencia
INSERT INTO employees VALUES (106, 'Fiona', 'Lee', 'flee@example.com',
'555-6789', TO_DATE('2017-05-12','YYYY-MM-DD'), 'SA_REP', 5200, NULL,
NULL, 10);
```

```
INSERT INTO employees VALUES (107, 'George', 'Miller', 'gmiller@example.com',
'555-7890', TO_DATE('2016-08-25','YYYY-MM-DD'), 'MK_MAN', 6200, NULL,
NULL, 20);
INSERT INTO employees VALUES (108, 'Hana', 'Yamamoto',
'hyamamoto@example.com', '555-8901', TO_DATE('2015-12-30','YYYY-MM-DD'),
'IT_PROG', 5800, NULL, NULL, 30);
INSERT INTO employees VALUES (109, 'Ivan', 'Petrov', 'ipetrov@example.com',
'555-9012', TO_DATE('2014-04-18','YYYY-MM-DD'), 'HR_ASS', 3100, NULL,
NULL, 40);
INSERT INTO employees VALUES (110, 'Julia', 'Martinez',
'jmartinez@example.com', '555-0123', TO_DATE('2013-07-07','YYYY-MM-DD'),
'FI_ANAL', 7300, NULL, NULL, 50);
```

2.7. Job_History

```
INSERT INTO job_history VALUES (101,
TO_DATE('2018-01-15','YYYY-MM-DD'),
TO_DATE('2019-01-15','YYYY-MM-DD'), 'SA_REP', 10);
INSERT INTO job_history VALUES (101,
TO_DATE('2019-01-16','YYYY-MM-DD'),
TO_DATE('2020-01-15','YYYY-MM-DD'), 'MK_MAN', 20);
INSERT INTO job_history VALUES (102,
TO_DATE('2019-03-10','YYYY-MM-DD'),
TO_DATE('2020-03-10','YYYY-MM-DD'), 'MK_MAN', 20);
INSERT INTO job_history VALUES (103,
TO_DATE('2020-06-01','YYYY-MM-DD'),
TO_DATE('2021-06-01','YYYY-MM-DD'), 'IT_PROG', 30);
```

```
INSERT INTO job_history VALUES (106,
TO_DATE('2017-05-12','YYYY-MM-DD'),
TO_DATE('2018-05-12','YYYY-MM-DD'), 'SA_REP', 10);
INSERT INTO job_history VALUES (106,
TO_DATE('2018-05-13','YYYY-MM-DD'),
TO_DATE('2019-05-12','YYYY-MM-DD'), 'MK_MAN', 20);
INSERT INTO job_history VALUES (106,
TO_DATE('2019-05-13','YYYY-MM-DD'),
TO_DATE('2020-05-12','YYYY-MM-DD'), 'HR_ASS', 40);
INSERT INTO job_history VALUES (107,
TO_DATE('2016-08-25','YYYY-MM-DD'),
TO_DATE('2017-08-25','YYYY-MM-DD'), 'MK_MAN', 20);
INSERT INTO job_history VALUES (107,
TO_DATE('2017-08-26','YYYY-MM-DD'),
TO_DATE('2018-08-25','YYYY-MM-DD'), 'FI_ANAL', 50);
INSERT INTO job_history VALUES (107,
TO_DATE('2018-08-26','YYYY-MM-DD'),
TO_DATE('2019-08-25','YYYY-MM-DD'), 'SA_REP', 10);
```

2.8. Horario

```
INSERT INTO horario VALUES ('MONDAY', 'Mañana',
TO_DATE('08:00','HH24:MI'), TO_DATE('12:00','HH24:MI'));
INSERT INTO horario VALUES ('MONDAY', 'Tarde',
TO_DATE('13:00','HH24:MI'), TO_DATE('17:00','HH24:MI'));
INSERT INTO horario VALUES ('TUESDAY', 'Mañana',
TO_DATE('08:00','HH24:MI'), TO_DATE('12:00','HH24:MI'));
```

```
INSERT INTO horario VALUES ('TUESDAY', 'Tarde',
TO_DATE('13:00','HH24:MI'), TO_DATE('17:00','HH24:MI'));
INSERT INTO horario VALUES ('WEDNESDAY', 'Mañana',
TO_DATE('08:00','HH24:MI'), TO_DATE('12:00','HH24:MI'));
INSERT INTO horario VALUES ('WEDNESDAY', 'Tarde',
TO_DATE('13:00','HH24:MI'), TO_DATE('17:00','HH24:MI'));
INSERT INTO horario VALUES ('THURSDAY', 'Mañana',
TO_DATE('08:00','HH24:MI'), TO_DATE('12:00','HH24:MI'));
INSERT INTO horario VALUES ('THURSDAY', 'Tarde',
TO_DATE('13:00','HH24:MI'), TO_DATE('17:00','HH24:MI'));
INSERT INTO horario VALUES ('FRIDAY', 'Mañana',
TO_DATE('08:00','HH24:MI'), TO_DATE('12:00','HH24:MI'));
INSERT INTO horario VALUES ('FRIDAY', 'Tarde', TO_DATE('13:00','HH24:MI'),
TO_DATE('17:00','HH24:MI'));
```

2.9. Empleado_horario

```
INSERT INTO empleado_horario VALUES ('MONDAY', 'Mañana', 101);
INSERT INTO empleado_horario VALUES ('MONDAY', 'Tarde', 101);
INSERT INTO empleado_horario VALUES ('TUESDAY', 'Mañana', 102);
INSERT INTO empleado_horario VALUES ('TUESDAY', 'Tarde', 102);
INSERT INTO empleado_horario VALUES ('WEDNESDAY', 'Mañana', 103);
INSERT INTO empleado_horario VALUES ('WEDNESDAY', 'Tarde', 103);
INSERT INTO empleado_horario VALUES ('THURSDAY', 'Mañana', 104);
INSERT INTO empleado_horario VALUES ('THURSDAY', 'Tarde', 104);
INSERT INTO empleado_horario VALUES ('FRIDAY', 'Mañana', 105);
INSERT INTO empleado_horario VALUES ('FRIDAY', 'Tarde', 105);
```

2.10. Asistencia_empleado

```
INSERT INTO asistencia_empleado VALUES (101, 'MONDAY',
TO_DATE('2025-10-20','YYYY-MM-DD'), TO_DATE('08:00','HH24:MI'),
TO_DATE('12:00','HH24:MI'));
INSERT INTO asistencia_empleado VALUES (101, 'MONDAY',
TO_DATE('2025-10-20','YYYY-MM-DD'), TO_DATE('13:00','HH24:MI'),
TO_DATE('17:00','HH24:MI'));
INSERT INTO asistencia_empleado VALUES (102, 'TUESDAY',
TO_DATE('2025-10-21','YYYY-MM-DD'), TO_DATE('08:00','HH24:MI'),
TO_DATE('12:00','HH24:MI'));
INSERT INTO asistencia_empleado VALUES (102, 'TUESDAY',
TO_DATE('2025-10-21','YYYY-MM-DD'), TO_DATE('13:00','HH24:MI'),
TO_DATE('17:00','HH24:MI'));
INSERT INTO asistencia_empleado VALUES (103, 'WEDNESDAY',
TO_DATE('2025-10-22','YYYY-MM-DD'), TO_DATE('08:00','HH24:MI'),
TO_DATE('12:00','HH24:MI'));
INSERT INTO asistencia_empleado VALUES (103, 'WEDNESDAY',
TO_DATE('2025-10-22','YYYY-MM-DD'), TO_DATE('13:00','HH24:MI'),
TO_DATE('17:00','HH24:MI'));
INSERT INTO asistencia_empleado VALUES (104, 'THURSDAY',
TO_DATE('2025-10-23','YYYY-MM-DD'), TO_DATE('08:00','HH24:MI'),
TO_DATE('12:00','HH24:MI'));
INSERT INTO asistencia_empleado VALUES (104, 'THURSDAY',
TO_DATE('2025-10-23','YYYY-MM-DD'), TO_DATE('13:00','HH24:MI'),
TO_DATE('17:00','HH24:MI'));
```

```
INSERT INTO asistencia_empleado VALUES (105, 'FRIDAY',
TO_DATE('2025-10-24','YYYY-MM-DD'), TO_DATE('08:00','HH24:MI'),
TO_DATE('12:00','HH24:MI'));
INSERT INTO asistencia_empleado VALUES (105, 'FRIDAY',
TO_DATE('2025-10-24','YYYY-MM-DD'), TO_DATE('13:00','HH24:MI'),
TO_DATE('17:00','HH24:MI'));
```

2.11. Capacitacion

```
INSERT INTO capacitacion VALUES (1, 'Seguridad Informática', 8, 'Curso sobre
ciberseguridad básica');
INSERT INTO capacitacion VALUES (2, 'Gestión de Proyectos', 10, 'Taller de
metodologías ágiles');
INSERT INTO capacitacion VALUES (3, 'Excel Avanzado', 6, 'Curso práctico de
hojas de cálculo');
INSERT INTO capacitacion VALUES (4, 'Comunicación Efectiva', 4, 'Seminario de
habilidades blandas');
INSERT INTO capacitacion VALUES (5, 'Bases de Datos Oracle', 12, 'Capacitación
técnica en SQL y PL/SQL');
INSERT INTO capacitacion VALUES (6, 'Inteligencia Artificial', 10, 'Introducción a
modelos de ML');
INSERT INTO capacitacion VALUES (7, 'Docker y DevOps', 8, 'Contenedores y
automatización');
INSERT INTO capacitacion VALUES (8, 'Finanzas Personales', 5, 'Educación
financiera básica');
INSERT INTO capacitacion VALUES (9, 'Marketing Digital', 7, 'Estrategias de redes
sociales');
```

INSERT INTO capacitacion VALUES (10, 'Diseño UX/UI', 9, 'Principios de diseño centrado en el usuario');

2.12. Empleado_capacitacion

INSERT INTO empleado_capacitacion VALUES (101, 1);

INSERT INTO empleado_capacitacion VALUES (101, 3);

INSERT INTO empleado_capacitacion VALUES (102, 2);

INSERT INTO empleado_capacitacion VALUES (102, 4);

INSERT INTO empleado_capacitacion VALUES (103, 5);

INSERT INTO empleado_capacitacion VALUES (103, 6);

INSERT INTO empleado_capacitacion VALUES (104, 7);

INSERT INTO empleado_capacitacion VALUES (104, 8);

INSERT INTO empleado_capacitacion VALUES (105, 9);

INSERT INTO empleado_capacitacion VALUES (105, 10);

INSERT INTO empleado_capacitacion VALUES (106, 1);

INSERT INTO empleado_capacitacion VALUES (106, 2);

INSERT INTO empleado_capacitacion VALUES (107, 3);

INSERT INTO empleado_capacitacion VALUES (108, 4);

INSERT INTO empleado_capacitacion VALUES (109, 5);

INSERT INTO empleado_capacitacion VALUES (110, 6);

INSERT INTO empleado_capacitacion VALUES (110, 7);

2.13. Creación de paquete con procedimientos y funciones CRUD

-- Especificación del paquete

CREATE OR REPLACE PACKAGE employee_pkg AS

```
  PROCEDURE crear_empleado(p_id NUMBER, p_fname VARCHAR2, p_lname
VARCHAR2, p_email VARCHAR2, p_hire DATE, p_job VARCHAR2, p_sal
NUMBER);
  PROCEDURE leer_empleado(p_id NUMBER);
  PROCEDURE actualizar_salario(p_id NUMBER, p_salario NUMBER);
  PROCEDURE eliminar_empleado(p_id NUMBER);
END employee_pkg;
/

-- Cuerpo del paquete
CREATE OR REPLACE PACKAGE BODY employee_pkg AS
  PROCEDURE crear_empleado(p_id NUMBER, p_fname VARCHAR2, p_lname
VARCHAR2, p_email VARCHAR2, p_hire DATE, p_job VARCHAR2, p_sal
NUMBER) IS
  BEGIN
    INSERT INTO employees (employee_id, first_name, last_name, email, hire_date,
job_id, salary)
    VALUES (p_id, p_fname, p_lname, p_email, p_hire, p_job, p_sal);
  END;

  PROCEDURE leer_empleado(p_id NUMBER) IS
    v_fname employees.first_name%TYPE;
    v_lname employees.last_name%TYPE;
  BEGIN
    SELECT first_name, last_name INTO v_fname, v_lname
    FROM employees WHERE employee_id = p_id;
```

```
  DBMS_OUTPUT.PUT_LINE('Empleado: ' || v_fname || ' ' || v_lname);
 END;

 PROCEDURE actualizar_salario(p_id NUMBER, p_salario NUMBER) IS
 BEGIN
  UPDATE employees SET salary = p_salario WHERE employee_id = p_id;
 END;

 PROCEDURE eliminar_empleado(p_id NUMBER) IS
 BEGIN
  DELETE FROM employees WHERE employee_id = p_id;
 END;
END employee_pkg;
/
```

2.13.1. Procedimento empleados con más rotaciones

```
CREATE OR REPLACE PROCEDURE empleados_mas_rotaciones IS
BEGIN
 FOR r IN (
  SELECT e.employee_id, e.last_name, e.first_name, e.job_id,
       j.job_title,
       COUNT(*) AS cambios
  FROM job_history jh
  JOIN employees e ON e.employee_id = jh.employee_id
  JOIN jobs j ON j.job_id = e.job_id
  GROUP BY e.employee_id, e.last_name, e.first_name, e.job_id, j.job_title
```

```
    ORDER BY cambios DESC
    FETCH FIRST 4 ROWS ONLY
  ) LOOP
    DBMS_OUTPUT.PUT_LINE(r.employee_id || ' - ' || r.last_name || ', ' ||
r.first_name || ' - ' || r.job_id || ' - ' || r.job_title || ' - Cambios: ' || r.cambios);
  END LOOP;
END;
/
```

2.13.2. Promedio de contrataciones por mes

```
CREATE OR REPLACE FUNCTION promedio_contrataciones_mes
RETURN NUMBER IS
  v_total NUMBER := 0;
BEGIN
  FOR r IN (
    SELECT TO_CHAR(hire_date, 'Month') AS mes,
          ROUND(COUNT(*) / COUNT(DISTINCT EXTRACT(YEAR
FROM hire_date)), 2) AS promedio
    FROM employees
    GROUP BY TO_CHAR(hire_date, 'Month')
  ) LOOP
    DBMS_OUTPUT.PUT_LINE(r.mes || ': ' || r.promedio);
    v_total := v_total + 1;
  END LOOP;
  RETURN v_total;
END;
```

```
/
```

2.13.3. Gostos y estadísticas por región

```
CREATE OR REPLACE PROCEDURE resumen_regional IS
BEGIN
  FOR r IN (
    SELECT rg.region_name,
           SUM(e.salary) AS total_salarios,
           COUNT(e.employee_id) AS total_empleados,
           MIN(e.hire_date) AS fecha_antigua
    FROM employees e
    JOIN departments d ON e.department_id = d.department_id
    JOIN locations l ON d.location_id = l.location_id
    JOIN countries c ON l.country_id = c.country_id
    JOIN regions rg ON c.region_id = rg.region_id
    GROUP BY rg.region_name
  ) LOOP
    DBMS_OUTPUT.PUT_LINE('Región: ' || r.region_name || ' | Total
salarios: ' || r.total_salarios || ' | Empleados: ' || r.total_empleados || ' | Antiguo: '
|| TO_CHAR(r.fecha_antigua, 'DD-MM-YYYY'));
  END LOOP;
END;
/
```

2.13.4. Tiempo de servicios y vacaciones

```
CREATE OR REPLACE FUNCTION calcular_vacaciones RETURN
NUMBER IS
  v_total_monto NUMBER := 0;
BEGIN
  FOR r IN (
    SELECT employee_id, first_name, last_name,
          MONTHS_BETWEEN(SYSDATE, hire_date)/12 AS anios,
          ROUND(MONTHS_BETWEEN(SYSDATE, hire_date)/12) AS
meses_vacaciones
    FROM employees
  ) LOOP
    DBMS_OUTPUT.PUT_LINE(r.employee_id || ' - ' || r.first_name || ' ' ||
r.last_name || ' | Años: ' || ROUND(r.anios,2) || ' | Vacaciones: ' ||
r.meses_vacaciones || ' meses');
    v_total_monto := v_total_monto + r.meses_vacaciones;
  END LOOP;
  RETURN v_total_monto;
END;
/
```

2.13.5. Horas laboradas en un mes

```
CREATE OR REPLACE FUNCTION horas_laboradas(p_emp_id
NUMBER, p_mes NUMBER, p_anio NUMBER) RETURN NUMBER IS
  v_horas NUMBER := 0;
BEGIN
  SELECT SUM(
```

```
    (hora_termino_real - hora_inicio_real) * 24
  )
  INTO v_horas
  FROM asistencia_empleado
  WHERE empleado_id = p_emp_id
    AND EXTRACT(MONTH FROM fecha) = p_mes
    AND EXTRACT(YEAR FROM fecha) = p_anio;

  RETURN NVL(v_horas, 0);
END;
/
```

2.13.6. Horas de falta en un mes

```
CREATE OR REPLACE FUNCTION horas_faltantes(p_emp_id NUMBER,
p_mes NUMBER, p_anio NUMBER) RETURN NUMBER IS
  v_horas_teoricas NUMBER := 0;
  v_horas_real NUMBER := 0;
BEGIN
  SELECT SUM((h.hora_termino - h.hora_inicio) * 24)
  INTO v_horas_teoricas
  FROM empleado_horario eh
  JOIN horario h ON eh.dia_semana = h.dia_semana AND eh.turno = h.turno
  WHERE eh.empleado_id = p_emp_id;

  v_horas_real := horas_laboradas(p_emp_id, p_mes, p_anio);
```

```
  RETURN GREATEST(v_horas_teoricas - v_horas_real, 0);
END;
/
```

2.13.7. Calculo de sueldo por mes

```
CREATE OR REPLACE PROCEDURE reporte_sueldo_mes(p_mes
NUMBER, p_anio NUMBER) IS
  v_salario_base NUMBER;
  v_horas_trabajadas NUMBER;
  v_horas_faltantes NUMBER;
BEGIN
  FOR r IN (
    SELECT employee_id, first_name, last_name, salary
    FROM employees
  ) LOOP
    v_salario_base := r.salary;
    v_horas_trabajadas := horas_laboradas(r.employee_id, p_mes, p_anio);
    v_horas_faltantes := horas_faltantes(r.employee_id, p_mes, p_anio);

    DBMS_OUTPUT.PUT_LINE(r.first_name || ' ' || r.last_name || ' | Sueldo
ajustado: ' ||
      ROUND(v_salario_base * (v_horas_trabajadas / (v_horas_trabajadas +
v_horas_faltantes)), 2));
  END LOOP;
END;
/
```

2.13.8. Horas totales de capacitación por empleado

```
CREATE OR REPLACE FUNCTION horas_capacitacion_empleado
RETURN NUMBER IS
  v_total_empleados NUMBER := 0;
BEGIN
  FOR r IN (
    SELECT ec.empleado_id,
          e.first_name || ' ' || e.last_name AS nombre,
          SUM(c.horas) AS total_horas
    FROM empleado_capacitacion ec
    JOIN capacitacion c ON ec.cod_capacitacion = c.cod_capacitacion
    JOIN employees e ON ec.empleado_id = e.employee_id
    GROUP BY ec.empleado_id, e.first_name, e.last_name
  ) LOOP
    DBMS_OUTPUT.PUT_LINE(r.nombre || ' | Total horas: ' || r.total_horas);
    v_total_empleados := v_total_empleados + 1;
  END LOOP;
  RETURN v_total_empleados;
END;
/
```

2.13.9. Listado de capacitaciones y horas por empleado

```
CREATE OR REPLACE PROCEDURE reporte_capacitaciones IS
BEGIN
  FOR r IN (
    SELECT c.nombre AS nombre_capacitacion,
```

```
        e.first_name || ' ' || e.last_name AS empleado,
        SUM(c.horas) AS total_horas
      FROM empleado_capacitacion ec
      JOIN capacitacion c ON ec.cod_capacitacion = c.cod_capacitacion
      JOIN employees e ON ec.empleado_id = e.employee_id
      GROUP BY c.nombre, e.first_name, e.last_name
      ORDER BY total_horas DESC
    ) LOOP
      DBMS_OUTPUT.PUT_LINE('Capacitación: ' || r.nombre_capacitacion || ' |
  Empleado: ' || r.empleado || ' | Horas: ' || r.total_horas);
    END LOOP;
  END;
  /
```

3. Trigger validación de asistencia

```
CREATE OR REPLACE TRIGGER validar_asistencia
BEFORE INSERT ON asistencia_empleado
FOR EACH ROW
DECLARE
  v_dia VARCHAR2(3);
  v_valido BOOLEAN := FALSE;
BEGIN
  -- Validar que el día de la fecha coincida con el campo dia_semana
  v_dia := TO_CHAR(:NEW.fecha, 'DY', 'NLS_DATE_LANGUAGE=ENGLISH');
  IF TRIM(UPPER(v_dia)) != TRIM(UPPER(SUBSTR(:NEW.dia_semana, 1, 3))) THEN
```

```
    RAISE_APPLICATION_ERROR(-20001, 'La fecha no corresponde al día de la semana
declarada.');
  END IF;

  -- Validar que exista al menos una coincidencia de horario asignado
  FOR r IN (
    SELECT h.hora_inicio, h.hora_termino
    FROM empleado_horario eh
    JOIN horario h ON eh.dia_semana = h.dia_semana AND eh.turno = h.turno
    WHERE eh.empleado_id = :NEW.empleado_id AND eh.dia_semana = :NEW.dia_semana
  ) LOOP
    IF :NEW.hora_inicio_real = r.hora_inicio AND :NEW.hora_termino_real = r.hora_termino
THEN
      v_valido := TRUE;
      EXIT;
    END IF;
  END LOOP;

  IF NOT v_valido THEN
    RAISE_APPLICATION_ERROR(-20002, 'Las horas reales no coinciden con ningún
horario asignado.');
  END IF;
END;
/
```

4. Trigger validación de salario según puesto

```
CREATE OR REPLACE TRIGGER validar_rango_salarial
BEFORE INSERT OR UPDATE ON employees
FOR EACH ROW
DECLARE
  v_min NUMBER;
  v_max NUMBER;
BEGIN
  SELECT min_salary, max_salary INTO v_min, v_max
  FROM jobs WHERE job_id = :NEW.job_id;

  IF :NEW.salary < v_min OR :NEW.salary > v_max THEN
    RAISE_APPLICATION_ERROR(-20003, 'El salario está fuera del rango permitido para el
puesto.');
  END IF;
END;
/
```

5. Trigger restricción de ingreso fuera de horario

```
CREATE OR REPLACE TRIGGER marcar_inasistencia
BEFORE INSERT ON asistencia_empleado
FOR EACH ROW
DECLARE
  v_hora_inicio DATE;
BEGIN
  SELECT h.hora_inicio INTO v_hora_inicio
  FROM empleado_horario eh
```

```
  JOIN horario h ON eh.dia_semana = h.dia_semana AND eh.turno = h.turno
  WHERE eh.empleado_id = :NEW.empleado_id AND eh.dia_semana = :NEW.dia_semana;

  IF ABS(:NEW.hora_inicio_real - v_hora_inicio) > (30/1440) THEN
    INSERT INTO asistencia_empleado (
      empleado_id, dia_semana, fecha, hora_inicio_real, hora_termino_real
    ) VALUES (
      :NEW.empleado_id, :NEW.dia_semana, :NEW.fecha, NULL, NULL
    );
    RAISE_APPLICATION_ERROR(-20004, 'Ingreso fuera de rango permitido. Se registró
inasistencia.');
  END IF;
END;
/
```

6. Script