

```
CREATE TABLE REGIONS (  
    region_id INT,  
    region_name VARCHAR2(50),  
    CONSTRAINT country_id_pk PRIMARY KEY (country_id)  
);
```

```
CREATE TABLE COUNTRIES (  
    country_id CHAR(2),  
    country_name VARCHAR(50),  
    region_id INT,  
    CONSTRAINT country_id_pk PRIMARY KEY (country_id),  
    CONSTRAINT region_id_in_countries FOREIGN KEY (region_id) REFERENCES REGIONS(region_id)  
);
```

```
CREATE TABLE LOCATIONS (  
    location_id INT,  
    street_address VARCHAR(100),  
    postal_code VARCHAR(20),  
    city VARCHAR(50),  
    state_province VARCHAR(50),  
    country_id CHAR(2),  
    CONSTRAINT location_id_pk PRIMARY KEY (location_id),  
    CONSTRAINT country_id_in_locations FOREIGN KEY (country_id) REFERENCES  
COUNTRIES(country_id)  
);
```

```
CREATE TABLE DEPARTMENTS (  
    department_id INT,
```

```
department_name VARCHAR2(100),  
  
manager_id INT,  
  
location_id INT,  
  
CONSTRAINT department_id_pk PRIMARY KEY (department_id),  
  
CONSTRAINT location_id_in_departments FOREIGN KEY (location_id) REFERENCES LOCATIONS  
(location_id)  
  
);
```

```
CREATE TABLE EMPLOYEES (  
  
    employee_id INT,  
  
    first_name VARCHAR2(50),  
  
    last_name VARCHAR2(50) NOT NULL,  
  
    email VARCHAR2(100) UNIQUE NOT NULL,  
  
    phone_number VARCHAR2(20),  
  
    hire_date DATE NOT NULL,  
  
    job_id INT,  
  
    salary NUMBER(10,2) NOT NULL,  
  
    commission_pct NUMBER(5,2),  
  
    manager_id INT,  
  
    department_id INT,  
  
    CONSTRAINT employee_id_pk PRIMARY KEY (employee_id),  
  
    CONSTRAINT manager_id_in_employees FOREIGN KEY (manager_id) REFERENCES EMPLOYEES  
(employee_id),  
  
    CONSTRAINT department_id_in_employees FOREIGN KEY (department_id) REFERENCES  
DEPARTMENTS (department_id)  
  
);
```

```
ALTER TABLE DEPARTMENTS ADD CONSTRAINT manager_id_in_departements FOREIGN KEY  
(manager_id) REFERENCES EMPLOYEES (employee_id);
```

```
CREATE TABLE JOB_HISTORY (  
    employee_id INT,  
    start_date DATE NOT NULL,  
    end_date DATE NOT NULL,  
    job_id INT NOT NULL,  
    department_id INT NOT NULL,  
    CONSTRAINT start_date_pk PRIMARY KEY (start_date),  
    CONSTRAINT department_id_in_job_history FOREIGN KEY (department_id) REFERENCES  
DEPARTMENTS (department_id),  
    CONSTRAINT employee_id_in_job_history FOREIGN KEY (employee_id) REFERENCES EMPLOYEES  
(employee_id)  
);
```

```
CREATE TABLE JOBS (  
    job_id INT,  
    job_title VARCHAR2(255) NOT NULL,  
    min_salary NUMBER(10,2) NOT NULL,  
    max_salary NUMBER(10,2) NOT NULL,  
    CONSTRAINT job_id_pk PRIMARY KEY (job_id)  
);
```

```
ALTER TABLE JOB_HISTORY ADD CONSTRAINT job_id_in_job_history FOREIGN KEY (job_id)  
REFERENCES JOBS (job_id);
```

```
ALTER TABLE EMPLOYEES ADD CONSTRAINT job_id_in_employees FOREIGN KEY (job_id) REFERENCES  
JOBS (job_id);
```

```
ALTER TABLE JOBS ADD CONSTRAINT chk_salary CHECK (max_salary >= min_salary + 2000);
```

```
INSERT INTO JOBS (job_id, job_title, min_salary, max_salary) VALUES (JOBS_SEQ.NEXTVAL, 'Developer', 3000, 7000);
```

```
INSERT INTO JOBS (job_id, job_title, min_salary, max_salary) VALUES (JOBS_SEQ.NEXTVAL, 'Manager', 5000, 10000);
```

```
INSERT INTO JOBS (job_id, job_title, min_salary, max_salary) VALUES (JOBS_SEQ.NEXTVAL, 'Analyst', 4000, 9000);
```

```
INSERT INTO JOBS (job_id, job_title, min_salary, max_salary) VALUES (JOBS_SEQ.NEXTVAL, 'Support', 2500, 6000);
```

```
SELECT * FROM JOBS;
```

```
INSERT INTO EMPLOYEES (employee_id, first_name, last_name, email, phone_number, hire_date, job_id, salary, commission_pct, manager_id, department_id) VALUES
```

```
(EMPLOYEES_SEQ.NEXTVAL, 'Jan', 'Kowalski', 'jan.k@example.com', '123456789', SYSDATE, 1, 5000, NULL, NULL, NULL);
```

```
INSERT INTO EMPLOYEES (employee_id, first_name, last_name, email, phone_number, hire_date, job_id, salary, commission_pct, manager_id, department_id) VALUES
```

```
(EMPLOYEES_SEQ.NEXTVAL, 'Anna', 'Nowak', 'anna.n@example.com', '987654321', SYSDATE, 2, 7000, 5, 3, NULL);
```

```
INSERT INTO EMPLOYEES (employee_id, first_name, last_name, email, phone_number, hire_date, job_id, salary, commission_pct, manager_id, department_id) VALUES
```

```
(EMPLOYEES_SEQ.NEXTVAL, 'Piotr', 'Zieliński', 'piotr.z@example.com', '555666777', SYSDATE, 3, 6000, 3, 3, NULL);
```

```
INSERT INTO EMPLOYEES (employee_id, first_name, last_name, email, phone_number, hire_date, job_id, salary, commission_pct, manager_id, department_id) VALUES
```

```
(EMPLOYEES_SEQ.NEXTVAL, 'Maria', 'Wiśniewska', 'maria.w@example.com', '444333222', SYSDATE, 4, 4500, NULL, 2, NULL);
```

```
SELECT * FROM EMPLOYEES;
```

```
UPDATE EMPLOYEES SET manager_id = 1 WHERE employee_id IN (2, 3);
```

```
UPDATE JOBS SET min_salary = min_salary + 500, max_salary = max_salary + 500 WHERE  
LOWER(job_title) LIKE '%b%' OR LOWER(job_title) LIKE '%s%';
```

```
SELECT DISTINCT job_id FROM EMPLOYEES WHERE job_id IN (SELECT job_id FROM jobs WHERE  
max_salary > 9000);
```

```
UPDATE EMPLOYEES SET manager_id = NULL WHERE manager_id IN (SELECT employee_id FROM  
employees WHERE job_id IN (2, 3));
```

```
DELETE FROM EMPLOYEES WHERE job_id IN (SELECT job_id FROM JOBS WHERE max_salary > 9000);
```

```
DELETE FROM JOBS WHERE max_salary > 9000;
```

```
DROP TABLE EMPLOYEES;
```

```
FLASHBACK TABLE EMPLOYEES TO BEFORE DROP;
```