```
-- DROP SCRIPT

-- Drop indexes
BEGIN
    EXECUTE IMMEDIATE 'DROP INDEX idx_salary_employee';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE != -1418 THEN
            RAISE;
        END IF;
END;
/

BEGIN
    EXECUTE IMMEDIATE 'DROP INDEX idx_payment_employee';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE != -1418 THEN
            RAISE;
        END IF;
END;
/

BEGIN
    EXECUTE IMMEDIATE 'DROP INDEX idx_tax_employee';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE != -1418 THEN
            RAISE;
        END IF;
END;
/

BEGIN
```

```
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE != -1418 THEN
            RAISE;
        END IF;
END;
/

-- Drop triggers
BEGIN
    EXECUTE IMMEDIATE 'DROP TRIGGER TRG_BEFORE_INSERT_DESIGNATION';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE != -4080 THEN
            RAISE;
        END IF;
END;
/

BEGIN
    EXECUTE IMMEDIATE 'DROP TRIGGER TRG_BEFORE_INSERT_EMPLOYEE';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE != -4080 THEN
            RAISE;
        END IF;
END;
/

BEGIN
    EXECUTE IMMEDIATE 'DROP TRIGGER TRG_BEFORE_INSERT_SALARY';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE != -4080 THEN
```

```
        END IF;
END;
/

BEGIN
    EXECUTE IMMEDIATE 'DROP TRIGGER TRG_BEFORE_INSERT_PAYMENT';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE != -4080 THEN
            RAISE;
        END IF;
END;
/

BEGIN
    EXECUTE IMMEDIATE 'DROP TRIGGER TRG_BEFORE_INSERT_TAX';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE != -4080 THEN
            RAISE;
        END IF;
END;
/

BEGIN
    EXECUTE IMMEDIATE 'DROP TRIGGER TRG_BEFORE_INSERT_DEDUCTION';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE != -4080 THEN
            RAISE;
        END IF;
END;
/
```

```
BEGIN
    EXECUTE IMMEDIATE 'DROP TABLE DEDUCTION';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE != -942 THEN
            RAISE;
        END IF;
END;
/

BEGIN
    EXECUTE IMMEDIATE 'DROP TABLE TAX';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE != -942 THEN
            RAISE;
        END IF;
END;
/

BEGIN
    EXECUTE IMMEDIATE 'DROP TABLE PAYMENT';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE != -942 THEN
            RAISE;
        END IF;
END;
/

BEGIN
    EXECUTE IMMEDIATE 'DROP TABLE SALARY';
EXCEPTION
    WHEN OTHERS THEN
```

```
            RAISE;
        END IF;
END;
/

BEGIN
    EXECUTE IMMEDIATE 'DROP TABLE EMPLOYEE';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE != -942 THEN
            RAISE;
        END IF;
END;
/

BEGIN
    EXECUTE IMMEDIATE 'DROP TABLE DESIGNATION';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE != -942 THEN
            RAISE;
        END IF;
END;
/

-- Drop sequences
BEGIN
    EXECUTE IMMEDIATE 'DROP SEQUENCE SEQ_DEDUCTION';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE != -2289 THEN
            RAISE;
        END IF;
END;
```

```
BEGIN
    EXECUTE IMMEDIATE 'DROP SEQUENCE SEQ_TAX';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE != -2289 THEN
            RAISE;
        END IF;
END;
/

BEGIN
    EXECUTE IMMEDIATE 'DROP SEQUENCE SEQ_PAYMENT';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE != -2289 THEN
            RAISE;
        END IF;
END;
/

BEGIN
    EXECUTE IMMEDIATE 'DROP SEQUENCE SEQ_SALARY';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE != -2289 THEN
            RAISE;
        END IF;
END;
/

BEGIN
    EXECUTE IMMEDIATE 'DROP SEQUENCE SEQ_EMPLOYEE';
EXCEPTION
```

```
        IF SQLCODE != -2289 THEN
            RAISE;
        END IF;
END;
/

BEGIN
    EXECUTE IMMEDIATE 'DROP SEQUENCE SEQ_DESIGNATION';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE != -2289 THEN
            RAISE;
        END IF;
END;
/

-- END OF DROP SCRIPT



-- Sequences for auto-generating primary keys
CREATE SEQUENCE SEQ_DESIGNATION START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE SEQ_EMPLOYEE START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE SEQ_SALARY START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE SEQ_PAYMENT START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE SEQ_TAX START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE SEQ_DEDUCTION START WITH 1 INCREMENT BY 1;

-- DESIGNATION table
CREATE TABLE DESIGNATION (
    DESIGNATION_ID NUMBER PRIMARY KEY,
    TITLE VARCHAR2(100) NOT NULL UNIQUE
);
```

```
CREATE TABLE EMPLOYEE (
    EMPLOYEE_ID NUMBER PRIMARY KEY,
    NAME VARCHAR2(100) NOT NULL,
    DESIGNATION_ID NUMBER UNIQUE REFERENCES DESIGNATION(DESIGNATION
);

-- SALARY table
CREATE TABLE SALARY (
    SALARY_ID NUMBER PRIMARY KEY,
    EMPLOYEE_ID NUMBER UNIQUE REFERENCES EMPLOYEE(EMPLOYEE_ID),
    AMOUNT NUMBER(10,2) NOT NULL CHECK (AMOUNT >= 0)
);

-- Check if the index already exists, if not, then create it
DECLARE
    index_exists NUMBER;
BEGIN
    SELECT COUNT(*) INTO index_exists FROM user_indexes WHERE index
    IF index_exists = 0 THEN
        EXECUTE IMMEDIATE 'CREATE INDEX idx_salary_employee ON SAL
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE = -1408 THEN
            NULL; -- Ignore the error if the column is already inc
        ELSE
            RAISE; -- Propagate any other errors
        END IF;
END;
/


-- PAYMENT table
CREATE TABLE PAYMENT (
```

```
    EMPLOYEE_ID NUMBER REFERENCES EMPLOYEE(EMPLOYEE_ID),
    AMOUNT NUMBER(10,2) NOT NULL CHECK (AMOUNT >= 0),
    DATE_RECEIVED DATE DEFAULT SYSDATE
);

-- TAX table
CREATE TABLE TAX (
    TAX_ID NUMBER PRIMARY KEY,
    EMPLOYEE_ID NUMBER REFERENCES EMPLOYEE(EMPLOYEE_ID),
    TAX_AMOUNT NUMBER(10,2) NOT NULL CHECK (TAX_AMOUNT >= 0)
);

-- DEDUCTION table
CREATE TABLE DEDUCTION (
    DEDUCTION_ID NUMBER PRIMARY KEY,
    EMPLOYEE_ID NUMBER REFERENCES EMPLOYEE(EMPLOYEE_ID),
    DEDUCTION_AMOUNT NUMBER(10,2) NOT NULL CHECK (DEDUCTION_AMOUNT
    REASON VARCHAR2(200)
);

-- Triggers for auto-populating IDs from sequences upon insert

-- For DESIGNATION
CREATE OR REPLACE TRIGGER TRG_BEFORE_INSERT_DESIGNATION
BEFORE INSERT ON DESIGNATION
FOR EACH ROW
BEGIN
    :NEW.DESIGNATION_ID := SEQ_DESIGNATION.NEXTVAL;
END;
/

-- For EMPLOYEE
CREATE OR REPLACE TRIGGER TRG_BEFORE_INSERT_EMPLOYEE
BEFORE INSERT ON EMPLOYEE
```

```
BEGIN
    :NEW.EMPLOYEE_ID := SEQ_EMPLOYEE.NEXTVAL;
END;
/

-- For SALARY
CREATE OR REPLACE TRIGGER TRG_BEFORE_INSERT_SALARY
BEFORE INSERT ON SALARY
FOR EACH ROW
BEGIN
    :NEW.SALARY_ID := SEQ_SALARY.NEXTVAL;
END;
/

-- For PAYMENT
CREATE OR REPLACE TRIGGER TRG_BEFORE_INSERT_PAYMENT
BEFORE INSERT ON PAYMENT
FOR EACH ROW
BEGIN
    :NEW.PAYMENT_ID := SEQ_PAYMENT.NEXTVAL;
END;
/

-- For TAX
CREATE OR REPLACE TRIGGER TRG_BEFORE_INSERT_TAX
BEFORE INSERT ON TAX
FOR EACH ROW
BEGIN
    :NEW.TAX_ID := SEQ_TAX.NEXTVAL;
END;
/

-- For DEDUCTION
CREATE OR REPLACE TRIGGER TRG_BEFORE_INSERT_DEDUCTION
```

```
FOR EACH ROW
BEGIN
    :NEW.DEDUCTION_ID := SEQ_DEDUCTION.NEXTVAL;
END;
/

-- Indexes
CREATE INDEX idx_salary_employee ON SALARY(EMPLOYEE_ID);
CREATE INDEX idx_payment_employee ON PAYMENT(EMPLOYEE_ID);
CREATE INDEX idx_tax_employee ON TAX(EMPLOYEE_ID);
CREATE INDEX idx_deduction_employee ON DEDUCTION(EMPLOYEE_ID);
```