1. Department Table

The table will be called `department`, with `dept_name` as the primary key, and the `budget` attribute will have a check constraint to ensure it is positive.

```
CREATE TABLE department (
    dept_name VARCHAR(50) NOT NULL,
    building VARCHAR(50),
    budget DECIMAL(10, 2) NOT NULL,
    CONSTRAINT pk_department PRIMARY KEY (dept_name),
    CONSTRAINT chk_budget_positive CHECK (budget > 0)
);
```

2. Course Table

The table will be called `course`, with `course_id` as the primary key, `dept_name` as a foreign key referencing the `department` table, and a check constraint on `credits` to ensure it is greater than or equal to 1.

```
CREATE TABLE course (
    course_id INT NOT NULL,
    title VARCHAR(100),
    dept_name VARCHAR(50) NOT NULL,
    credits INT NOT NULL,
    CONSTRAINT pk_course PRIMARY KEY (course_id),
    CONSTRAINT fk_course_dept FOREIGN KEY (dept_name) REFERENCES
department(dept_name) ON DELETE CASCADE,
    CONSTRAINT chk_credits_positive CHECK (credits >= 1)
);
```

Primary Key (pk_department): Ensures `dept_name` uniquely identifies each row in the `department` table.
Check Constraint (chk_budget_positive): Enforces that `budget` must be positive in the `department` table.

Foreign Key (fk_course_dept): Links `dept_name` in the `course` table to `dept_name` in the `department` table, with `ON DELETE CASCADE` to automatically delete courses if the corresponding department is deleted.
Primary Key (pk_course): Ensures `course_id` uniquely identifies each row in the `course` table.
Check Constraint (chk_credits_positive): Ensures `credits` must be at least 1 in the `course` table.

2. Schema Modification

1. ALTER TABLE department ADD head_of_department VARCHAR(50);

2. ALTER TABLE department DROP COLUMN building;

3. ALTER TABLE course MODIFY credits DECIMAL(4, 2);

4. ALTER TABLE course RENAME COLUMN title TO course_title;

5. ALTER TABLE department RENAME TO dept;

6. ALTER TABLE course ADD CONSTRAINT fk_course_dept FOREIGN KEY (dept_name) REFERENCES department(dept_name);

7. ALTER TABLE course DROP CONSTRAINT fk_course_dept;

8. SELECT * FROM user_cons_columns WHERE TABLE_NAME = 'DEPARTMENT';

9. DROP TABLE course;

3. Manipulating Data (DML)

1. SELECT dept_name, budget FROM department WHERE *budget > 50000;*

2. INSERT INTO department (dept_name, building, budget) VALUES ('Computer Science', 'Building A', 100000);

3. DELETE FROM department WHERE dept_name = 'History';

4. UPDATE department SET budget = budget * 1.1 WHERE dept_name = 'Mathematics';

5. SELECT * FROM instructor, department;

6. SELECT * FROM instructor, department WHERE instructor.dept_name = department.dept_name;

7. SELECT * FROM instructor NATURAL JOIN department;