Here's an outline to create those scripts:

1. Data Types:

Define tables with different data types for columns. For example:

```
CREATE TABLE  STUDENT (
    id INT,
    name VARCHAR(50),
    age INT,
    birth_date DATE,
    salary DECIMAL(10, 2), -- Decimal data type
    is_employed BOOLEAN,  -- Boolean data type
    description TEXT,      -- Text data type
    created_at TIMESTAMP,  -- Timestamp data type
    image BLOB,            -- Binary Large Object (BLOB) data type
    rating FLOAT,          -- Floating-point data type
    json_data JSON,        -- JSON data type (if supported by your DBMS)
    status ENUM('Active', 'Inactive', 'Pending')  -- ENUM data type
    -- Add other columns with various data types
);
```

2. Constraints (including CHECK):

Apply constraints to enforce rules on columns:

```
CREATE TABLE STUDENT (
    id INT PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    age INT,
    CONSTRAINT age_check CHECK (age >= 18),
    CONSTRAINT name_length_check CHECK (LENGTH(name) > 3 AND LENGTH(name) <= 50),
    CONSTRAINT positive_id_check CHECK (id > 0)

);
```

3. Index:

Create indexes to improve query performance:

```
CREATE INDEX idx_name ON ExampleTable (name);
```

4. Views:

Create a view based on a query:

```
CREATE VIEW SimpleView AS
SELECT id, name FROM ExampleTable;

CREATE VIEW ComplexView AS
SELECT t.id, t.name, c.category_name
FROM ExampleTable t
JOIN CategoryTable c ON t.category_id = c.id;

CREATE MATERIALIZED VIEW MaterializedView AS
SELECT id, name FROM ExampleTable;


CREATE VIEW UpdatableView AS
SELECT id, name FROM ExampleTable WHERE condition_column = 'value' WITH CHECK OPTION;

CREATE VIEW IndexedView WITH SCHEMABINDING AS
SELECT id, name FROM ExampleTable;
CREATE UNIQUE CLUSTERED INDEX idx_IndexedView ON IndexedView(id);
```

5. Synonyms:

Create a synonym for a table or view:

```
CREATE SYNONYM ExampleSynonym FOR ExampleTable;
```

6. Sequence:

Create a sequence for generating unique numbers:

```
CREATE SEQUENCE ExampleSequence START WITH 1 INCREMENT BY 1;
```

7. Roles and Privileges:

Define roles and grant privileges to users:

```
CREATE ROLE ExampleRole;
GRANT SELECT, INSERT ON ExampleTable TO ExampleRole;
```

8. Triggers:

Create triggers to perform actions on specific events:

```
CREATE TRIGGER BeforeInsertStudent
BEFORE INSERT ON Student
FOR EACH ROW
BEGIN
    -- Trigger logic here
    -- For instance, to set a default value for a column:
    IF NEW.column_name IS NULL THEN
        SET NEW.column_name = 'default_value';
    END IF;
    -- You can add your specific logic or operations here.
END;
```