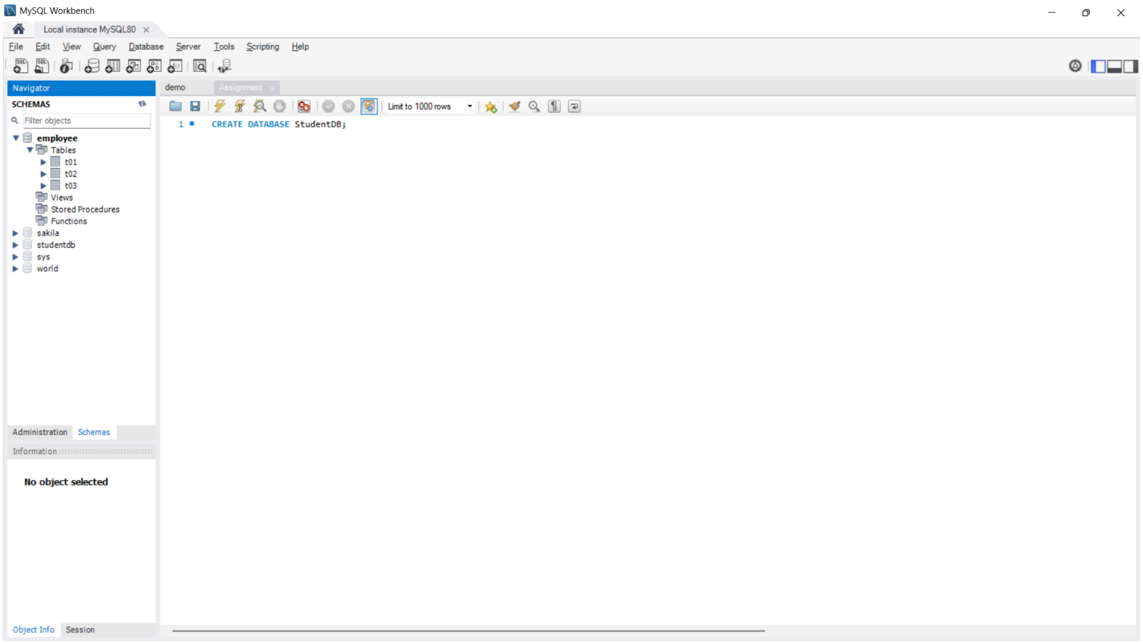
# Week 1 - Assignments

| | |
|---|---|
| Created by | Dakshil Gorasiya |

## Assignment 1

```
CREATE DATABASE StudentDB;
```



```
CREATE DATABASE StudentDB;
```

## Assignment 2

```
-- Courses
CREATE TABLE T02
(
        T02F01 INT PRIMARY KEY,  -- course_id
    T02F02 VARCHAR(50) NOT NULL, -- course_nae
    T02F03 INT NOT NULL -- duration
);

-- Students
CREATE TABLE T01
(
        T01F01 INT PRIMARY KEY,  -- student_id
    T01F02 VARCHAR(50) NOT NULL, -- name
    T01F03 INT NOT NULL, -- age
    T01F04 ENUM("MALE", "FEMALE", "OTHER"), -- gender
    T01F05 INT, -- course_id
    CONSTRAINT T01F05_FK FOREIGN KEY (T01F05) REFERENCES T02(T02F01) ON DELETE RE
STRICT
);

-- Marks
CREATE TABLE T03
(
        T03F01 INT PRIMARY KEY, -- mark_id
    T03F02 INT, -- student_id
    T03F03 VARCHAR(50) NOT NULL, -- subject
    T03F04 DECIMAL(5, 2) NOT NULL,
    CONSTRAINT T03F02_FK FOREIGN KEY (T03F02) REFERENCES T01(T01F01) ON DELETE CA
SCADE
);

ALTER TABLE T01 ADD COLUMN T01F06 VARCHAR(100) NOT NULL UNIQUE;  -- email

DROP TABLE T03;
```

# Assignment 3

```
INSERT INTO T02 (T02F01, T02F02, T02F03) VALUES
(101, 'Computer Science', 4),
(102, 'Business Administration', 3),
(103, 'Data Analytics', 1),
(104, 'MACHINE LEARNING', 2),
(105, 'WEB DEVELOPMENT', 2);

INSERT INTO T01 (T01F01, T01F02, T01F03, T01F04, T01F05, T01F06) VALUES
(1, 'Alice Johnson', 21, 'Female', 101, 'alice@gmail.com'),
(2, 'Bob Smith', 22, 'Male', 101, 'bob@gmail.com'),
(3, 'Charlie Brown', 20, 'Male', 102, 'charlie@gmail.com'),
(4, 'Diana Prince', 23, 'Female', 103, 'diana@gmail.om'),
(5, 'Eve Adams', 21, 'Female', 102, 'eve@gmail.com');

INSERT INTO T03 (T03F01, T03F02, T03F03, T03F04) VALUES
(1, 1, 'OS', 85),
(2, 2, 'ACCOUNTS', 91),
(3, 3, 'PYTHON', 76),
(4, 4, 'TENSORFLOW', 95),
(5, 5, 'COA', 88);

UPDATE T01 SET T01F05 = 104 WHERE T01F01 = 1;

DELETE FROM T01 WHERE T01F01 = 1;
```

# Assignment 4

```
-- 1. Fetch all students above age 20
SELECT
    T01F02,
    T01F03
FROM
    T01
WHERE
    T01F03 > 20;

-- 2. Fetch all students ordered by name alphabetically.
SELECT
    T01F02
FROM
    T01
ORDER BY
    T01F02;

-- 3. Show total number of students enrolled in each course using GROUP BY.
SELECT
    T02F01,
  T02F02,
  COUNT(T01F05)
FROM
    T01
  INNER JOIN T02 ON T02F01 = T01F05
GROUP BY
    T01F05;

-- 4. Show courses that have more than 2 students using HAVING.
SELECT
    T02F01,
  T02F02,
  COUNT(T01F05)
FROM
    T01
  INNER JOIN T02 ON T02F01 = T01F05
GROUP BY
```

```
    T01F05
HAVING
    COUNT(T01F05) > 2;
```

## Assignment 5

```
-- 1. Display students with their enrolled course names using INNER JOIN.
SELECT
    T01F01,
    T01F02,
    T02F02
FROM
    T01
    INNER JOIN T02 ON T01F05 = T02F01;

-- 2. Display all students even if they are not enrolled in any course (LEFT JOIN).
SELECT
    T01F01,
    T01F02,
    COALESCE(T02F02, 'Not Enrolled')
FROM
    T01
    LEFT JOIN T02 ON T01F05 = T02F01;

-- 3. Display all courses and their students (RIGHT JOIN).
SELECT
    T02F01,
    T02F02,
    COUNT(T01F05)
FROM
    T01
    RIGHT JOIN T02 ON T01F05 = T02F01
GROUP BY
    T02F01;

-- 4. Find highest, lowest, and average marks per subject.
SELECT
    T03F03,
    MAX(T03F04),
    MIN(T03F04),
    AVG(T03F04)
FROM
    T03
GROUP BY
    T03F03;
```

```
-- 5. Count how many male and female students exist.
SELECT
    T01F04,
    COUNT(T01F04)
FROM
    T01
GROUP BY
    T01F04;
```