

Lab No.2

Exploring SQL Plus Commands

1. Create a table **Student** based on the following chart:

Column	Data type	Constraints
Student Id	Number (6)	Primary Key
Last Name	Varchar2(15)	Not NULL
First Name	Varchar2(15)	Not NULL
gender	Char(3)	

Confirm and validate the creation of the new table.

Answer.

```
postgres=# CREATE TABLE Student (  
postgres(# Student_Id SERIAL PRIMARY KEY,  
postgres(# Last_Name VARCHAR(15) NOT NULL,  
postgres(# First_Name VARCHAR(15) NOT NULL,  
postgres(# Gender CHAR(3));  
CREATE TABLE
```

```
postgres=# \d Student  
  
          Table "public.student"  
  Column      |      Type      | Collation | Nullable |      Default        
-----  
student_id    | integer        |           | not null | nextval('student_student_id_seq'::regclass)  
last_name     | character varying(15) |           | not null |   
first_name    | character varying(15) |           | not null |   
gender        | character(3)    |           |          |   
Indexes:  
"student_pkey" PRIMARY KEY, btree (student_id)
```

2. Create a table **Dept** based on the following chart:

Column	Data type	Constraints
Dept_Code	Char (3)	Not NULL
Dept_Name	Varchar2(20)	Not NULL

Confirm and validate the creation of the new table.

Answer.

```
postgres=# CREATE TABLE Dept (  
postgres(# Dept_Code  
postgres(# CHAR(3) NOT NULL,  
postgres(# Dept_Name VARCHAR(20) NOT NULL);
```

```
postgres=# \d Dept
```

Column	Type	Collation	Nullable	Default
dept_code	character(3)		not null	
dept_name	character varying(20)		not null	

3. Add a new column **Location** to **Dept** table which has data type Char(7). Confirm and validate the modification of the table.

Answer.

```
postgres=# ALTER TABLE Dept
postgres=# ADD COLUMN Location CHAR(7);
ALTER TABLE
```

```
postgres=# \d Dept
```

Column	Type	Collation	Nullable	Default
dept_code	character(3)		not null	
dept_name	character varying(20)		not null	
location	character(7)			

4. Delete the column **Last_Name** from **Student**. Confirm and validate the modification of the table.

Answer.

```
postgres=# ALTER TABLE Student
postgres=# DROP COLUMN Last_Name;
ALTER TABLE
postgres=# \d Student
```

Column	Type	Collation	Nullable	Default
student_id	integer		not null	nextval('student_student_id_seq'::regclass)
first_name	character varying(15)		not null	
gender	character(3)			

Indexes:

```
"student_pkey" PRIMARY KEY, btree (student_id)
```

5. Increase **Last_Name** column to **25** characters long. Save the SQL statement as *ex5.sql*. Confirm and validate the modification of the table.

Answer.

```
postgres=# ALTER TABLE Student
postgres=# ALTER COLUMN Last_Name TYPE VARCHAR(15);
ALTER TABLE
```

```
postgres=# \echo 'ALTER TABLE Student ALTER COLUMN Last_Name TYPE VARCHAR(25);' > ex5.sql
ALTER TABLE Student ALTER COLUMN Last_Name TYPE VARCHAR(25); > ex5.sql
postgres=# \d Student
```

Column	Type	Collation	Nullable	Default
student_id	integer		not null	nextval('student_student_id_seq'::regclass)
first_name	character varying(15)		not null	
gender	character(3)			
last_name	character varying(15)			

Indexes:

"student_pkey" PRIMARY KEY, btree (student_id)

6. Create 2 more tables that you think are necessary for student data management.

Answer.

```
postgres=# CREATE TABLE Course (
postgres=# Course_Id SERIAL PRIMARY KEY,
postgres=# Course_Name VARCHAR(50) NOT NULL,
postgres=# Course_Code VARCHAR(10) UNIQUE NOT NULL);
CREATE TABLE
```

```
postgres=# Enrollment_Id SERIAL PRIMARY KEY,
postgres=# Student_Id INT NOT NULL,
postgres=# Course_Id INT NOT NULL,
postgres=# Enrollment_Date DATE,
postgres=# FOREIGN KEY (Student_Id) REFERENCES Student(Student_Id), FOREIGN KEY (Course_Id) REFERENCES Course(Course_Id));
CREATE TABLE
```

7. Insert values in all the tables.

Answer.

```
postgres=# -- Insert values into Student table
postgres=# INSERT INTO Student (Last_Name, First_Name, Gender)
postgres=# VALUES ('Doe', 'John', 'M'),
postgres=#          ('Smith', 'Alice', 'F');
```

```
postgres=# INSERT INTO Course (Course_Name, Course_Code)
postgres=# VALUES ('Mathematics', 'MATH101'),
postgres=#          ('English Composition', 'ENG101');
```

```
postgres=# -- Insert values into Enrollment table
postgres=# INSERT INTO Enrollment (Student_Id, Course_Id, Enrollment_Date)
postgres=# VALUES (1, 1, '2024-01-15'),
postgres=#          (2, 2, '2024-01-20');
```

8. Select specific data from the tables

Answer.

```
postgres=# SELECT First_Name, Last_Name FROM Student;
 first_name | last_name
-----+-----
    John    |    Doe
    Alice    |   Smith
(2 rows)
```

```
postgres=# SELECT Course_Name FROM Course WHERE Course_Code = 'MATH101';
 course_name
-----
 Mathematics
(1 row)
```

```
postgres=# SELECT * FROM Enrollment WHERE Student_Id = 1;
 enrollment_id | student_id | course_id | enrollment_date
-----+-----+-----+-----
            1 |          1 |          1 | 2024-01-15
(1 row)
```

9. Select entire data from the tables

Answer.

```
postgres=# SELECT * FROM Student;
 student_id | first_name | gender | last_name
-----+-----+-----+-----
          1 | John      | M      | Doe
          2 | Alice     | F      | Smith
(2 rows)
```

```
postgres=# SELECT * FROM Course;
 course_id | course_name | course_code
-----+-----+-----
          1 | Mathematics | MATH101
          2 | English Composition | ENG101
(2 rows)
```

```
postgres=# SELECT * FROM Enrollment;
 enrollment_id | student_id | course_id | enrollment_date
-----+-----+-----+-----
              1 |          1 |          1 | 2024-01-15
              2 |          2 |          2 | 2024-01-20
(2 rows)
```

10. Truncate student table

Answer.

```
postgres=# TRUNCATE TABLE Student CASCADE;
NOTICE: truncate cascades to table "enrollment"
TRUNCATE TABLE
```

11. Add 2 new columns to dept table

Answer.

```
postgres=# ALTER TABLE Dept
postgres-# ADD COLUMN Chairperson VARCHAR(50),
postgres-# ADD COLUMN Budget DECIMAL(10,2);
ALTER TABLE
```

12. Delete a particular column from dept table.

Answer.

```
postgres=# ALTER TABLE Dept
postgres=# DROP COLUMN Budget;
ALTER TABLE
```

13. Create arithmetic calculations

Answer.

```
postgres=# SELECT Student_Id, COUNT(*) AS Total_Courses
postgres=# FROM Enrollment
postgres=# GROUP BY Student_Id;
 student_id | total_courses
-----+-----
```

14. Select distinct values from the column

Answer.

```
postgres=# SELECT DISTINCT Gender FROM Student;
gender
-----
 M
  F
(2 rows)
```

15. Concatenate columns and display

Answer.

```
postgres=# -- Concatenate First_Name and Last_Name columns and display as full name
postgres=# SELECT CONCAT(First_Name, ' ', Last_Name) AS Full_Name FROM Student;
 full_name
-----
 John Doe
 Alice Smith
 Michael Johnson
 Emily Williams
(4 rows)
```

16. Drop all tables.

Answer.

```
postgres=# DROP TABLE IF EXISTS Enrollment;
DROP TABLE
postgres=# DROP TABLE IF EXISTS Course;
DROP TABLE
postgres=# DROP TABLE IF EXISTS Student;
DROP TABLE
postgres=# DROP TABLE IF EXISTS Dept;
DROP TABLE
postgres=# \d
Did not find any relations.
postgres=# _
```