**TABLE CREATION AND MODIFICATION (FK)**

*--\_\_\_\_\_\_\_\_\_\_\_\_A\_\_\_\_\_\_\_\_\_\_\_\_--*

create table patient\_tb(

    PatientID **INT** **PRIMARY KEY**

   ,FirstName **VARCHAR**(50)

   ,LastName **VARCHAR**(50)

   ,Gender **CHAR**(1)

   ,BirthDate **DATE**

   ,Phone **VARCHAR**(15)

   ,Email **VARCHAR**(100) UNIQUE

);

*-- ADD CONSTRAIN PRIMARY KEY FOR PatientID , UNIQUE FOR EAMIL*

*-- 1: FIRST MAKE IT NULL*

ALTER TABLE patient\_tb

ALTER COLUMN PatientID **INT** NOT NULL;

*-- 2: AND THEN ADD CONSTRAIN PRIMARY KEY*

*--a*

ALTER TABLE patient\_tb

ADD **CONSTRAINT** Pk\_PatientId **PRIMARY KEY** (PatientID);

*--b*

ALTER TABLE patient\_tb

ADD **CONSTRAINT** Uk\_Email UNIQUE(Email);

*-- 3: DROP A CONSTRAIN*

ALTER TABLE patient\_tb

DROP **CONSTRAINT** Pk\_PatientId;

*--DROP TABLE patient\_tb*

*--\_\_\_\_\_\_\_\_\_\_\_\_B\_\_\_\_\_\_\_\_\_\_\_\_--*

create table department\_tb(

    DepartmentID **INT** **PRIMARY KEY**,

    DepartmentName **VARCHAR**(100),

);

*--\_\_\_\_\_\_\_\_\_\_\_\_C\_\_\_\_\_\_\_\_\_\_\_\_--*

*--1*

create table doctor\_tb(

    DoctorID **INT** **PRIMARY KEY**,

    FirstName **VARCHAR**(50),

    LastName **VARCHAR**(50),

    Specialty **VARCHAR**(100),

    DepartmentID **INT** **FOREIGN KEY** **REFERENCES** department\_tb(DepartmentID)

);

*--2*

create table doctor1\_tb(

    DoctorID **INT** **PRIMARY KEY**,

    FirstName **VARCHAR**(50),

    LastName **VARCHAR**(50),

    Specialty **VARCHAR**(100),

    DepartmentID **INT** ,

**CONSTRAINT** FK\_DEP\_ID **FOREIGN KEY**(DepartmentID) **REFERENCES** department\_tb(DepartmentID)

);

DROP TABLE doctor1\_tb

*--3*

create table doctor2\_tb(

    DoctorID **INT** **PRIMARY KEY**,

    FirstName **VARCHAR**(50),

    LastName **VARCHAR**(50),

    Specialty **VARCHAR**(100),

    DepartmentID **INT**

);

ALTER TABLE doctor2\_tb

ADD **CONSTRAINT** FK\_DEP\_ID **FOREIGN KEY**(DepartmentID) **REFERENCES** department\_tb(DepartmentID)

DROP table doctor2\_tb

*--\_\_\_\_\_\_\_\_\_\_\_\_D\_\_\_\_\_\_\_\_\_\_\_\_--*

create table treatment\_tb(

    TreatmentID **INT** **PRIMARY KEY**,

    PatientID **INT** **FOREIGN KEY** **REFERENCES** patient\_tb(PatientID),

    DoctorID **INT**  **FOREIGN KEY** **REFERENCES** doctor\_tb(DoctorID),

    TreatmentDate **DATE**,

    Diagnosis **VARCHAR**(50),

    Treatment **VARCHAR**(100)

);

**ALTER**

*-- 1. Rename a table (SQL Server does not support ALTER TABLE for renaming tables)*

*-- Instead, use sp\_rename to rename the table from 'OldTableName' to 'NewTableName'*

EXEC sp\_rename 'OldTableName', 'NewTableName';

*-- 2. Rename a column in a table (SQL Server does not support ALTER TABLE for renaming columns)*

*-- Instead, use sp\_rename to rename 'OldColumnName' to 'NewColumnName' in 'TableName'*

EXEC sp\_rename 'TableName.OldColumnName', 'NewColumnName', 'COLUMN';

*-- 3. Modify a column's data type*

*-- This changes the data type of 'ColumnName' in 'TableName' to 'NewDataType'*

ALTER TABLE TableName ALTER COLUMN ColumnName NewDataType;

*-- Example: Change 'age' column in 'Students' table to BIGINT*

ALTER TABLE Students ALTER COLUMN age **BIGINT**;

*-- 4. Add a new column to a table*

*-- This adds a new column 'NewColumnName' with the specified data type to 'TableName'*

ALTER TABLE TableName ADD NewColumnName DataType;

*-- Example: Add a 'phone\_number' column of type VARCHAR(15) to 'Students'*

ALTER TABLE Students ADD phone\_number **VARCHAR**(15);

*-- 5. Drop an existing column from a table*

*-- This removes 'ColumnName' from 'TableName'*

ALTER TABLE TableName DROP COLUMN ColumnName;

*-- Example: Remove the 'phone\_number' column from 'Students'*

ALTER TABLE Students DROP COLUMN phone\_number;

**ORDER\_BY AND GROUP\_BY**

*-- 1. ORDER BY: Sorts the result set based on a column*

*-- Example: Retrieve students' names and ages, sorted by age in ascending order*

SELECT name, age

FROM Students

ORDER BY age ASC; *-- Default sorting is ASC (ascending)*

*-- Example: Retrieve students' names and ages, sorted by age in descending order*

SELECT name, age

FROM Students

ORDER BY age DESC; *-- Sorting from highest to lowest age*

*-- Example: Sort by multiple columns*

SELECT name, age, grade

FROM Students

ORDER BY grade DESC, age ASC; *-- First sorts by grade (high to low), then by age (low to high)*

*-- 2. GROUP BY: Groups rows that have the same values in specified columns*

*-- Example: Count the number of students in each grade*

SELECT grade, COUNT(\*) AS student\_count

FROM Students

GROUP BY grade; *-- Groups all records by grade and counts the number of students in each*

*-- Example: Find the average age of students in each grade*

SELECT grade, AVG(age) AS avg\_age

FROM Students

GROUP BY grade; *-- Groups students by grade and calculates the average age in each group*

*-- Example: Group by multiple columns (grade and city)*

SELECT grade, city, COUNT(\*) AS student\_count

FROM Students

GROUP BY grade, city; *-- Groups students by grade and city, then counts how many students are in each group*

*-- 3. Using ORDER BY with GROUP BY*

*-- Example: Count students per grade and order by the highest count*

SELECT grade, COUNT(\*) AS student\_count

FROM Students

GROUP BY grade

ORDER BY student\_count DESC; *-- Groups students by grade, counts them, then sorts by the highest count first*