

UNIT 5

Creating and Altering Database and Tables (SQL)

(6 Hours) — Database Management System (CACS255) · BCA II Year / IV Semester

⚠ Important Note — Please Read

These notes were prepared quickly from the course syllabus and standard textbook knowledge. **They may contain errors, typing mistakes or incomplete points.** Please take care: cross-check definitions, formulas, SQL syntax and examples with your class notes, the prescribed textbook and your teacher before relying on them for exams or practicals.

5.1 Introduction to SQL

SQL (Structured Query Language) is the standard language for communicating with a relational database. It was developed at IBM in the 1970s (first called SEQUEL) and is standardized by ANSI/ISO. SQL is **non-procedural**: you say *what* you want and the DBMS decides *how* to get it.

Category	Purpose	Commands
DDL	Define/modify structure	CREATE, ALTER, DROP, TRUNCATE
DML	Change data	INSERT, UPDATE, DELETE
DQL	Query data	SELECT
DCL	Control access	GRANT, REVOKE
TCL	Manage transactions	COMMIT, ROLLBACK, SAVEPOINT

Note: The syllabus topics (stored procedures, triggers, RETURN, output parameters) follow the Microsoft SQL Server (T-SQL) style, so the examples in Units 5–7 use T-SQL. MySQL/Oracle syntax differs slightly in places and is mentioned where important.

Common Data Types

Type	Description	Example
INT / SMALLINT / BIGINT	Whole numbers	Age INT
DECIMAL(p, s) / NUMERIC	Exact numbers; p = total digits, s = digits after the point	Salary DECIMAL(10,2)
FLOAT / REAL	Approximate numbers	Height FLOAT

Type	Description	Example
CHAR(n)	Fixed-length string	Gender CHAR(1)
VARCHAR(n)	Variable-length string up to n	Name VARCHAR(50)
DATE / DATETIME	Date / date and time	DOB DATE
BIT (T-SQL) / BOOLEAN	True/false	IsActive BIT
TEXT / VARCHAR(MAX)	Long text	Remarks TEXT

5.2 Creating and Altering Database

CREATE DATABASE

The simplest form only gives the name:

Basic form

```
CREATE DATABASE CollegeDB;
USE CollegeDB;           -- select the database to work in
```

In SQL Server you can give **arguments** for the data file (.mdf) and log file (.ldf):

With arguments (SQL Server)

```
CREATE DATABASE CollegeDB
ON PRIMARY
( NAME          = CollegeDB_data,
  FILENAME      = 'C:\SQLData\CollegeDB.mdf',
  SIZE          = 10MB,
  MAXSIZE       = 100MB,
  FILEGROWTH    = 5MB )
LOG ON
( NAME          = CollegeDB_log,
  FILENAME      = 'C:\SQLData\CollegeDB.ldf',
  SIZE          = 5MB,
  MAXSIZE       = 50MB,
  FILEGROWTH    = 10% );
```

Argument	Meaning
ON PRIMARY	Defines the primary file group holding the data file
NAME	Logical name of the file used inside SQL Server
FILENAME	Physical path of the file on disk
SIZE	Initial size of the file
MAXSIZE	Maximum size the file can grow to (or UNLIMITED)

Argument	Meaning
FILEGROWTH	Growth increment — in MB or in percent
LOG ON	Defines the transaction log file

*Note: MySQL form: **CREATE DATABASE IF NOT EXISTS CollegeDB CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;***

ALTER DATABASE

Common ALTER DATABASE operations

```
ALTER DATABASE CollegeDB MODIFY NAME = CollegeDB2;           -- rename
ALTER DATABASE CollegeDB2
ADD FILE ( NAME = CollegeDB_data2,
           FILENAME = 'C:\SQLData\CollegeDB2.ndf', SIZE = 5MB ); -- add a file
ALTER DATABASE CollegeDB2
MODIFY FILE ( NAME = CollegeDB_data, SIZE = 20MB );           -- change size
ALTER DATABASE CollegeDB2 SET READ_ONLY;                       -- change option
```

5.3 Creating Tables with Constraints

CREATE TABLE defines a table's name, its columns with data types, and constraints. A **constraint** is a rule that the DBMS enforces on the data.

General syntax

```
CREATE TABLE table_name (
    column1 datatype [column_constraint],
    column2 datatype [column_constraint],
    ...
    [table_constraint]
);
```

Constraint	Purpose
NOT NULL	Column cannot hold NULL.
UNIQUE	All values in the column must be different.
PRIMARY KEY	Uniquely identifies rows; = NOT NULL + UNIQUE; only one per table.
FOREIGN KEY	Links to the primary key of another table (referential integrity).
CHECK	Value must satisfy a condition, e.g. Age >= 16.
DEFAULT	Value used when none is supplied.
IDENTITY / AUTO_INCREMENT	Automatically generated sequence number (T-SQL: IDENTITY(1,1); MySQL: AUTO_INCREMENT).

Normal (simple) tables

Department table

```
CREATE TABLE Department (  
    DeptID      INT          PRIMARY KEY,  
    DeptName    VARCHAR(50)  NOT NULL UNIQUE  
);
```

Complex tables — multiple constraints and foreign keys

Student table with all main constraints

```
CREATE TABLE Student (  
    StudentID   INT IDENTITY(1,1) PRIMARY KEY,  
    Name        VARCHAR(60)  NOT NULL,  
    Age         INT          CHECK (Age >= 16),  
    Email       VARCHAR(80)  UNIQUE,  
    Country     VARCHAR(30)  DEFAULT 'Nepal',  
    DeptID      INT,  
    CONSTRAINT FK_Student_Dept FOREIGN KEY (DeptID)  
        REFERENCES Department (DeptID)  
        ON DELETE SET NULL  
        ON UPDATE CASCADE  
);
```

Junction table with composite primary key

```
CREATE TABLE Enrollment (  
    StudentID   INT,  
    CourseID    INT,  
    Grade       CHAR(2),  
    EnrollDate  DATE DEFAULT GETDATE(),  
    CONSTRAINT PK_Enroll PRIMARY KEY (StudentID, CourseID), -- composite key  
    CONSTRAINT FK_E_Student FOREIGN KEY (StudentID) REFERENCES Student (StudentID),  
    CONSTRAINT FK_E_Course  FOREIGN KEY (CourseID)  REFERENCES Course (CourseID),  
    CONSTRAINT CK_Grade CHECK (Grade IN ('A', 'B', 'C', 'D', 'F'))  
);
```

Referential action	What happens when the parent row is deleted/updated
NO ACTION / RESTRICT	Operation is rejected if child rows exist (default).
CASCADE	The change is applied to the child rows too (child rows deleted/updated).
SET NULL	Child foreign key is set to NULL.
SET DEFAULT	Child foreign key is set to its default value.

*Note: The parent table (Department, Course) must be created **before** the child table that references it.*

5.4 ALTER TABLE

ALTER TABLE changes the structure of an existing table — columns and constraints — without losing its data.

Common ALTER TABLE operations

```
-- Add a column
ALTER TABLE Student ADD Phone VARCHAR(15);
-- Change the data type / size of a column (T-SQL)
ALTER TABLE Student ALTER COLUMN Phone VARCHAR(20) NOT NULL;
-- MySQL: ALTER TABLE Student MODIFY Phone VARCHAR(20) NOT NULL;
-- Drop a column
ALTER TABLE Student DROP COLUMN Phone;
-- Add constraints
ALTER TABLE Student ADD CONSTRAINT UQ_Phone UNIQUE (Phone);
ALTER TABLE Student ADD CONSTRAINT CK_Age CHECK (Age BETWEEN 16 AND 60);
ALTER TABLE Student ADD CONSTRAINT DF_Country DEFAULT 'Nepal' FOR Country; -- T-SQL
ALTER TABLE Student ADD CONSTRAINT FK_Dept FOREIGN KEY (DeptID)
    REFERENCES Department (DeptID);
-- Drop a constraint
ALTER TABLE Student DROP CONSTRAINT CK_Age;
-- MySQL foreign key: ALTER TABLE Student DROP FOREIGN KEY FK_Dept;
-- Rename (T-SQL)
EXEC sp_rename 'Student.Phone', 'Mobile', 'COLUMN';
```

5.5 DROP Statement

DROP statements

```
DROP TABLE Enrollment; -- deletes structure + data
DROP TABLE IF EXISTS Enrollment; -- no error if it does not exist
DROP DATABASE CollegeDB; -- deletes the whole database
```

Command	Type	Removes	Can rollback?	Structure kept?
DELETE	DML	Selected rows (WHERE) or all rows	Yes (inside a transaction)	Yes
TRUNCATE	DDL	All rows quickly; resets identity	Usually no (T-SQL allows inside a transaction)	Yes
DROP	DDL	Table structure and all data	No	No

Note: A table that is referenced by a foreign key cannot be dropped until the child table (or the constraint) is dropped first.

Example 5.1 — Create a Library database table

Create table **Book** with BookID (primary key, auto number), Title (not null), Price (must be more than 0), Status (default 'Available').

Solution

See the code below.

```
CREATE TABLE Book (  
    BookID INT IDENTITY(1,1) PRIMARY KEY,  
    Title VARCHAR(100) NOT NULL,  
    Price DECIMAL(8,2) CHECK (Price > 0),  
    Status VARCHAR(15) DEFAULT 'Available'  
);
```

Key Takeaways — Unit 5

- ✓ SQL has DDL, DML, DQL, DCL and TCL parts; DDL creates and changes structures.
- ✓ CREATE DATABASE can take arguments such as file name, size, max size and file growth (SQL Server).
- ✓ Constraints — NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK, DEFAULT — keep data valid.
- ✓ Create parent tables before child tables; composite keys are declared at table level.
- ✓ ALTER TABLE adds/drops columns and constraints; DROP removes objects; TRUNCATE empties a table.
- ✓ Syntax varies among DBMSs (T-SQL vs MySQL) — always check your lab software.

— *End of Unit 5* —

Reminder: these notes may contain errors — verify with your textbook and teacher.