Department of Operational Research
University of Delhi

Practical File

Submitted for the course
205: Database Management System
& Visual Programming

SUBMITTED BY
Om Gupta
Exam Roll No.: 24236761066
761: Master of Operational Research
North Department
Section – A
Batch – 2024-26

SUBMITTED TO
Dr. Gurjeet Kaur
Assistant Professor
Department of Operational Research

Table of Contents

1. Database and DDL command without constraints.

```
-- Create Database
CREATE DATABASE OG

-- Use the Database
USE OG

-- Create Table for BookMyShow
CREATE TABLE bookmyshow (
show_id CHAR(4),
show_name VARCHAR(50),
show_desc TEXT,
show_date DATE,
show_time TIME,
is_full BIT,
show_count TINYINT,
price DECIMAL(15,2),
capacity BIGINT)

-- Insert Data into the Table
INSERT INTO bookmyshow VALUES
('1001', 'The Weeknd', 'After Hours Tour', '2025-07-15',
'20:00:00', 0, 2, 150.00, 1800),
('1002', 'Imagine Dragons', 'Mercury Tour', '2025-08-10',
'19:30:00', 0, 4, 175.00, 2200),
('1003', 'Ed Sheeran', 'Mathematics Tour', '2025-09-05',
'18:00:00', 1, 6, 250.00, 2500),
('1004', 'Taylor Swift', 'Eras Tour', '2025-10-20',
'20:00:00', 0, 7, 300.00, 3500),
('1005', 'Billie Eilish', 'Happier Than Ever', '2025-06-12',
'19:00:00', 1, 5, 200.00, 2000),
('1006', 'Arijit Singh', 'Live in Concert', '2025-05-22',
'20:30:00', 0, 3, 150.50, 2700),
('1007', 'Bruno Mars', '24K Magic', '2025-04-18', '21:00:00',
0, 4, 280.00, 3000),
('1008', 'Shawn Mendes', 'Wonder Tour', '2025-03-25',
'18:45:00', 1, 3, 220.75, 2500),
('1009', 'Maroon 5', 'Red Pill Blues', '2025-11-14',
'19:15:00', 0, 5, 210.00, 2900),
('1010', 'OneRepublic', 'Human Tour', '2025-08-20',
'20:45:00', 0, 6, 185.00, 2600),
```

```
('1011', 'Post Malone', 'Twelve Carat Tour', '2025-09-30',
'21:15:00', 1, 4, 270.50, 2400),
('1012', 'Dua Lipa', 'Future Nostalgia', '2025-12-10',
'20:00:00', 0, 3, 230.00, 3000),
('1013', 'Adele', '30 Tour', '2025-11-25', '19:30:00', 0, 4,
350.00, 3100),
('1014', 'Katy Perry', 'Smile Tour', '2025-10-05', '20:15:00',
1, 5, 190.50, 2800),
('1015', 'Justin Bieber', 'Justice Tour', '2025-09-18',
'18:30:00', 0, 6, 260.00, 3200),
('1016', 'Green Day', 'Hella Mega Tour', '2025-08-15',
'19:00:00', 1, 3, 220.00, 2700),
('1017', 'Linkin Park', 'Hybrid Theory Live', '2025-07-21',
'21:00:00', 0, 7, 275.00, 3300),
('1018', 'Red Hot Chili Peppers', 'Unlimited Love Tour',
'2025-06-28', '20:30:00', 0, 4, 280.00, 3400),
('1020', 'Coldplay', NULL, '2025-12-01', '09:00:00', 0, 3,
200.50, 2500),
('1019', 'Xyz', NULL, '2026-12-31', '09:30:00', 1, 5, 210.15,
3000)

-- Retrieve Data from the Table
SELECT * FROM bookmyshow
```

	show_id	show_name	show_desc	show_date	show_time	is_full	show_count	price	capacity
1	1001	The Weeknd	After Hours Tour	2025-07-15	20:00:00.0000000	0	2	150.00	1800
2	1002	Imagine Dragons	Mercury Tour	2025-08-10	19:30:00.0000000	0	4	175.00	2200
3	1003	Ed Sheeran	Mathematics Tour	2025-09-05	18:00:00.0000000	1	6	250.00	2500
4	1004	Taylor Swift	Eras Tour	2025-10-20	20:00:00.0000000	0	7	300.00	3500
5	1005	Billie Eilish	Happier Than Ever	2025-06-12	19:00:00.0000000	1	5	200.00	2000
6	1006	Arijit Singh	Live in Concert	2025-05-22	20:30:00.0000000	0	3	150.50	2700
7	1007	Bruno Mars	24K Magic	2025-04-18	21:00:00.0000000	0	4	280.00	3000
8	1008	Shawn Mendes	Wonder Tour	2025-03-25	18:45:00.0000000	1	3	220.75	2500
9	1009	Maroon 5	Red Pill Blues	2025-11-14	19:15:00.0000000	0	5	210.00	2900
10	1010	OneRepublic	Human Tour	2025-08-20	20:45:00.0000000	0	6	185.00	2600
11	1011	Post Malone	Twelve Carat Tour	2025-09-30	21:15:00.0000000	1	4	270.50	2400
12	1012	Dua Lipa	Future Nostalgia	2025-12-10	20:00:00.0000000	0	3	230.00	3000
13	1013	Adele	30 Tour	2025-11-25	19:30:00.0000000	0	4	350.00	3100
14	1014	Katy Perry	Smile Tour	2025-10-05	20:15:00.0000000	1	5	190.50	2800
15	1015	Justin Bieber	Justice Tour	2025-09-18	18:30:00.0000000	0	6	260.00	3200
16	1016	Green Day	Hella Mega Tour	2025-08-15	19:00:00.0000000	1	3	220.00	2700
17	1017	Linkin Park	Hybrid Theory Live	2025-07-21	21:00:00.0000000	0	7	275.00	3300
18	1018	Red Hot Chili P...	Unlimited Love T...	2025-06-28	20:30:00.0000000	0	4	280.00	3400
19	1020	Coldplay	NULL	2025-12-01	09:00:00.0000000	0	3	200.50	2500
20	1019	Xyz	NULL	2026-12-31	09:30:00.0000000	1	5	210.15	3000

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (78) | OG | 00:00:00 | 20 rows

2. DDL commands with constraints (without and with foreign keys)

```
-- Create Database
CREATE DATABASE university


-- Use the Database
USE university


-- Create Teachers Table
CREATE TABLE teachers (
     teacher_id INT PRIMARY KEY,
     course_id VARCHAR(10) UNIQUE,
     teacher_name VARCHAR(50)
)


-- Create Students Table
CREATE TABLE students (
     roll_no INT PRIMARY KEY,
     name VARCHAR(50),
     aadhaar_no CHAR(12) NOT NULL UNIQUE,
     cgpa_10 FLOAT,
     cgpa_12 FLOAT,
     course_id VARCHAR(10),
     teacher_id INT,
     admission_status VARCHAR(3) CHECK (admission_status IN
     ('yes', 'no')),
     max_cgpa FLOAT DEFAULT 10,
     CONSTRAINT chk_cgpa_10_range CHECK (cgpa_10 BETWEEN 0 AND
     10),
     CONSTRAINT chk_cgpa_12_range CHECK (cgpa_12 BETWEEN 0 AND
     10),
     FOREIGN KEY (course_id) REFERENCES teachers(course_id),
     FOREIGN KEY (teacher_id) REFERENCES teachers(teacher_id),
     CHECK ((admission_status = 'yes' AND cgpa_10 IS NOT NULL
     AND cgpa_12 IS NOT NULL) OR
     (admission_status = 'no' AND cgpa_10 IS NULL AND cgpa_12
     IS NULL))
);
```

```sql
-- Insert Data into Teachers Table
INSERT INTO teachers VALUES
(1, 'CS101', 'Rajesh Sharma'),
(2, 'ME102', 'Anil Kumar'),
(3, 'EC103', 'Priya Mehta'),
(4, 'BT104', 'Amitabh Joshi'),
(5, 'CE105', 'Neha Reddy'),
(6, 'IT106', 'Arun Mishra'),
(7, 'EE107', 'Pooja Patel'),
(8, 'CH108', 'Ravi Verma'),
(9, 'MA109', 'Sanjay Gupta'),
(10, 'PH110', 'Kavita Iyer');


-- Insert Data into Students Table
INSERT INTO students VALUES
(101, 'Aarav Kapoor', '123456789012', 9.2, 8.8, 'CS101', 1,
'yes', 10),
(102, 'Ishaan Malhotra', '223456789012', 8.1, 7.9, 'ME102', 2,
'yes', 10),
(103, 'Riya Sharma', '323456789012', NULL, NULL, 'EC103', 3,
'no', 10),
(104, 'Ananya Gupta', '423456789012', 7.5, 8.1, 'BT104', 4,
'yes', 10),
(105, 'Kabir Khanna', '523456789012', NULL, NULL, 'CE105', 5,
'no', 10),
(106, 'Devansh Nair', '623456789012', 9.0, 9.3, 'IT106', 6,
'yes', 10),
(107, 'Aryan Sinha', '723456789012', NULL, NULL, 'EE107', 7,
'no', 10),
(108, 'Tara Menon', '823456789012', 7.8, 8.0, 'CH108', 8,
'yes', 10),
(109, 'Neel Patel', '923456789012', NULL, NULL, 'MA109', 9,
'no', 10),
(110, 'Sanya Kapoor', '103456789012', 8.7, 9.1, 'PH110', 10,
'yes', 10);

-- View Data
SELECT * FROM teachers
SELECT * FROM students
```

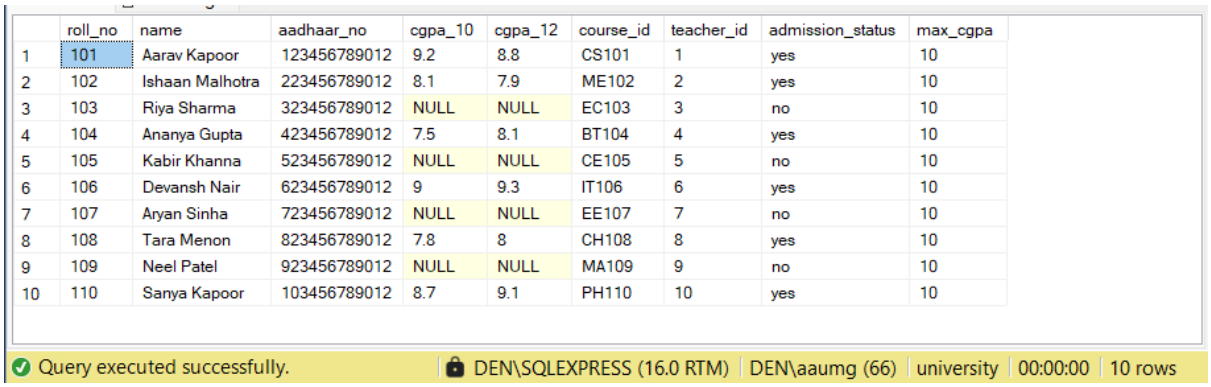
	teacher_id	course_id	teacher_name
1	1	CS101	Rajesh Sharma
2	2	ME102	Anil Kumar
3	3	EC103	Priya Mehta
4	4	BT104	Amitabh Joshi
5	5	CE105	Neha Reddy
6	6	IT106	Arun Mishra
7	7	EE107	Pooja Patel
8	8	CH108	Ravi Verma
9	9	MA109	Sanjay Gupta
10	10	PH110	Kavita Iyer

	roll_no	name	aadhaar_no	cgpa_10	cgpa_12	course_id	teacher_id	admission_status	max_cgpa
1	101	Aarav Kapoor	123456789012	9.2	8.8	CS101	1	yes	10
2	102	Ishaan Malhotra	223456789012	8.1	7.9	ME102	2	yes	10
3	103	Riya Sharma	323456789012	NULL	NULL	EC103	3	no	10
4	104	Ananya Gupta	423456789012	7.5	8.1	BT104	4	yes	10
5	105	Kabir Khanna	523456789012	NULL	NULL	CE105	5	no	10
6	106	Devansh Nair	623456789012	9	9.3	IT106	6	yes	10
7	107	Aryan Sinha	723456789012	NULL	NULL	EE107	7	no	10
8	108	Tara Menon	823456789012	7.8	8	CH108	8	yes	10
9	109	Neel Patel	923456789012	NULL	NULL	MA109	9	no	10
10	110	Sanya Kapoor	103456789012	8.7	9.1	PH110	10	yes	10

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (51) | university | 00:00:00 | 20 rows

3. Update DDL structures: ALTER, SET, UPDATE, etc.

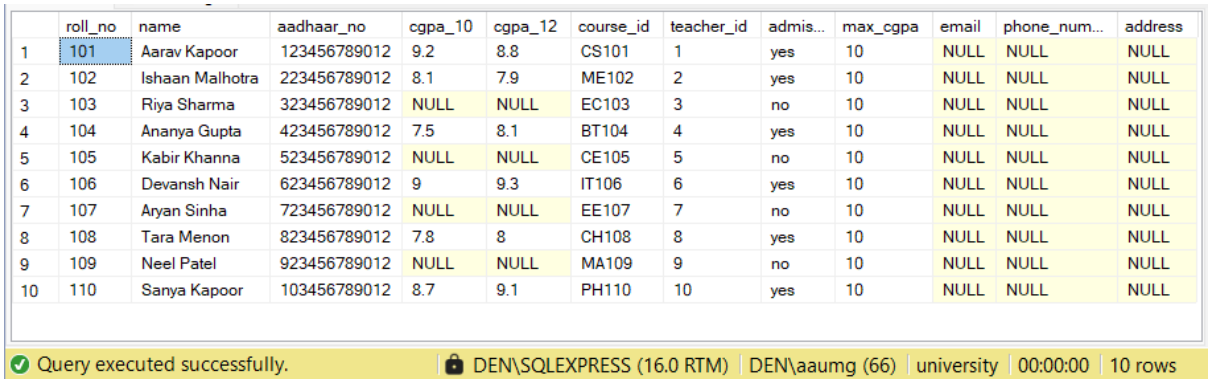
```
USE university
SELECT * FROM students
```



	roll_no	name	aadhaar_no	cgpa_10	cgpa_12	course_id	teacher_id	admission_status	max_cgpa
1	101	Aarav Kapoor	123456789012	9.2	8.8	CS101	1	yes	10
2	102	Ishaan Malhotra	223456789012	8.1	7.9	ME102	2	yes	10
3	103	Riya Sharma	323456789012	NULL	NULL	EC103	3	no	10
4	104	Ananya Gupta	423456789012	7.5	8.1	BT104	4	yes	10
5	105	Kabir Khanna	523456789012	NULL	NULL	CE105	5	no	10
6	106	Devansh Nair	623456789012	9	9.3	IT106	6	yes	10
7	107	Aryan Sinha	723456789012	NULL	NULL	EE107	7	no	10
8	108	Tara Menon	823456789012	7.8	8	CH108	8	yes	10
9	109	Neel Patel	923456789012	NULL	NULL	MA109	9	no	10
10	110	Sanya Kapoor	103456789012	8.7	9.1	PH110	10	yes	10

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (66) | university | 00:00:00 | 10 rows

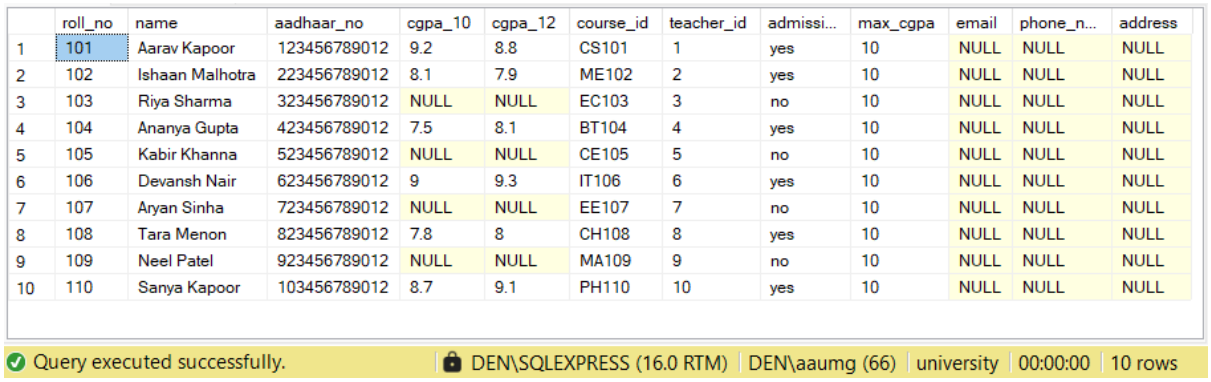
```
-- Add a new column to students table
ALTER TABLE students ADD email VARCHAR(50);
ALTER TABLE students ADD phone_number CHAR(10);
ALTER TABLE students ADD address VARCHAR(100);
SELECT * FROM students
```



	roll_no	name	aadhaar_no	cgpa_10	cgpa_12	course_id	teacher_id	admis...	max_cgpa	email	phone_num...	address
1	101	Aarav Kapoor	123456789012	9.2	8.8	CS101	1	yes	10	NULL	NULL	NULL
2	102	Ishaan Malhotra	223456789012	8.1	7.9	ME102	2	yes	10	NULL	NULL	NULL
3	103	Riya Sharma	323456789012	NULL	NULL	EC103	3	no	10	NULL	NULL	NULL
4	104	Ananya Gupta	423456789012	7.5	8.1	BT104	4	yes	10	NULL	NULL	NULL
5	105	Kabir Khanna	523456789012	NULL	NULL	CE105	5	no	10	NULL	NULL	NULL
6	106	Devansh Nair	623456789012	9	9.3	IT106	6	yes	10	NULL	NULL	NULL
7	107	Aryan Sinha	723456789012	NULL	NULL	EE107	7	no	10	NULL	NULL	NULL
8	108	Tara Menon	823456789012	7.8	8	CH108	8	yes	10	NULL	NULL	NULL
9	109	Neel Patel	923456789012	NULL	NULL	MA109	9	no	10	NULL	NULL	NULL
10	110	Sanya Kapoor	103456789012	8.7	9.1	PH110	10	yes	10	NULL	NULL	NULL

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (66) | university | 00:00:00 | 10 rows

```
-- Modify the datatype of teacher_name in teachers table
ALTER TABLE teachers ALTER COLUMN teacher_name VARCHAR(100)
SELECT * FROM students
```



	roll_no	name	aadhaar_no	cgpa_10	cgpa_12	course_id	teacher_id	admissi...	max_cgpa	email	phone_n...	address
1	101	Aarav Kapoor	123456789012	9.2	8.8	CS101	1	yes	10	NULL	NULL	NULL
2	102	Ishaan Malhotra	223456789012	8.1	7.9	ME102	2	yes	10	NULL	NULL	NULL
3	103	Riya Sharma	323456789012	NULL	NULL	EC103	3	no	10	NULL	NULL	NULL
4	104	Ananya Gupta	423456789012	7.5	8.1	BT104	4	yes	10	NULL	NULL	NULL
5	105	Kabir Khanna	523456789012	NULL	NULL	CE105	5	no	10	NULL	NULL	NULL
6	106	Devansh Nair	623456789012	9	9.3	IT106	6	yes	10	NULL	NULL	NULL
7	107	Aryan Sinha	723456789012	NULL	NULL	EE107	7	no	10	NULL	NULL	NULL
8	108	Tara Menon	823456789012	7.8	8	CH108	8	yes	10	NULL	NULL	NULL
9	109	Neel Patel	923456789012	NULL	NULL	MA109	9	no	10	NULL	NULL	NULL
10	110	Sanya Kapoor	103456789012	8.7	9.1	PH110	10	yes	10	NULL	NULL	NULL

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (66) | university | 00:00:00 | 10 rows

```sql
-- Update admission_status to 'yes' for student with roll_no
105
UPDATE students SET admission_status = 'yes', cgpa_10 = 7.0,
cgpa_12 = 7.5 WHERE roll_no = 105;
UPDATE students SET admission_status = 'no', cgpa_10 = NULL,
cgpa_12 = NULL WHERE roll_no = 107;
UPDATE students SET admission_status = 'yes', cgpa_10 = 8.2,
cgpa_12 = 8.5 WHERE roll_no = 103;
SELECT * FROM students
```

	roll_no	name	aadhaar_no	cgpa_10	cgpa_12	course_id	teacher_id	admissi...	max_cgpa	email	phone_nu...	address
1	101	Aarav Kapoor	123456789012	9.2	8.8	CS101	1	yes	10	NULL	NULL	NULL
2	102	Ishaan Malhotra	223456789012	8.1	7.9	ME102	2	yes	10	NULL	NULL	NULL
3	103	Riya Sharma	323456789012	8.2	8.5	EC103	3	yes	10	NULL	NULL	NULL
4	104	Ananya Gupta	423456789012	7.5	8.1	BT104	4	yes	10	NULL	NULL	NULL
5	105	Kabir Khanna	523456789012	7	7.5	CE105	5	yes	10	NULL	NULL	NULL
6	106	Devansh Nair	623456789012	9	9.3	IT106	6	yes	10	NULL	NULL	NULL
7	107	Aryan Sinha	723456789012	NULL	NULL	EE107	7	no	10	NULL	NULL	NULL
8	108	Tara Menon	823456789012	7.8	8	CH108	8	yes	10	NULL	NULL	NULL
9	109	Neel Patel	923456789012	NULL	NULL	MA109	9	no	10	NULL	NULL	NULL
10	110	Sanya Kapoor	103456789012	8.7	9.1	PH110	10	yes	10	NULL	NULL	NULL

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (66) | university | 00:00:00 | 10 rows

```sql
-- Modifying check condition on cgpa
ALTER TABLE students DROP CONSTRAINT chk_cgpa_10_range;
ALTER TABLE students ADD CONSTRAINT chk_cgpa_10_range CHECK
(cgpa_10 BETWEEN 0 AND 100);
ALTER TABLE students DROP CONSTRAINT chk_cgpa_12_range;
ALTER TABLE students ADD CONSTRAINT chk_cgpa_12_range CHECK
(cgpa_12 BETWEEN 0 AND 100);

-- Set max_cgpa, cgpas out of 100
UPDATE students SET max_cgpa = 100;
UPDATE students SET cgpa_10 = cgpa_10*10;
UPDATE students SET cgpa_12 = cgpa_12*10;
SELECT * FROM students
```

	roll_no	name	aadhaar_no	cgpa_10	cgpa_12	course_id	teacher_id	admissi...	max_cgpa	email	phone_num...	address
1	101	Aarav Kapoor	123456789012	92	88	CS101	1	yes	100	NULL	NULL	NULL
2	102	Ishaan Malhotra	223456789012	81	79	ME102	2	yes	100	NULL	NULL	NULL
3	103	Riya Sharma	323456789012	82	85	EC103	3	yes	100	NULL	NULL	NULL
4	104	Ananya Gupta	423456789012	75	81	BT104	4	yes	100	NULL	NULL	NULL
5	105	Kabir Khanna	523456789012	70	75	CE105	5	yes	100	NULL	NULL	NULL
6	106	Devansh Nair	623456789012	90	93	IT106	6	yes	100	NULL	NULL	NULL
7	107	Aryan Sinha	723456789012	NULL	NULL	EE107	7	no	100	NULL	NULL	NULL
8	108	Tara Menon	823456789012	78	80	CH108	8	yes	100	NULL	NULL	NULL
9	109	Neel Patel	923456789012	NULL	NULL	MA109	9	no	100	NULL	NULL	NULL
10	110	Sanya Kapoor	103456789012	87	91	PH110	10	yes	100	NULL	NULL	NULL

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (66) | university | 00:00:00 | 10 rows

```sql
-- Rename column 'student_name' to 'name' in students table
EXEC sp_rename 'students.student_name', 'name', 'COLUMN';
```

```
EXEC sp_rename 'teachers.teacher_name', 'name', 'COLUMN';
SELECT * FROM students
```

	roll_no	name	aadhaar_no	cgpa_10	cgpa_12	course_id	teacher_id	admission_status	max_cgpa	email	phone_number
1	101	Aarav Kapoor	123456789012	92	88	CS101	1	yes	100	NULL	NULL
2	102	Ishaan Malhotra	223456789012	81	79	ME102	2	yes	100	NULL	NULL
3	103	Riya Sharma	323456789012	82	85	EC103	3	yes	100	NULL	NULL
4	104	Ananya Gupta	423456789012	75	81	BT104	4	yes	100	NULL	NULL
5	105	Kabir Khanna	523456789012	70	75	CE105	5	yes	100	NULL	NULL
6	106	Devansh Nair	623456789012	90	93	IT106	6	yes	100	NULL	NULL
7	107	Aryan Sinha	723456789012	NULL	NULL	EE107	7	no	100	NULL	NULL
8	108	Tara Menon	823456789012	78	80	CH108	8	yes	100	NULL	NULL
9	109	Neel Patel	923456789012	NULL	NULL	MA109	9	no	100	NULL	NULL
10	110	Sanya Kapoor	103456789012	87	91	PH110	10	yes	100	NULL	NULL

Query completed with errors. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (66) | university | 00:00:00 | 10 rows

4. Steps to Import database and perform at least 15 queries using SELECT, *, DISTINCT, LITERAL SELECT, CASE END, FROM, WHERE and ORDER BY.

```
/*
Step: Import database and perform 15 queries using SELECT, *,
DISTINCT, LITERAL SELECT, CASE END, FROM, WHERE, and ORDER BY.
Database: university
Assumption: The database and tables (students, teachers) are
already created and populated.
*/

USE university;

-- 1. SELECT all columns from students table
SELECT * FROM students;
```

	roll_no	name	aadhaar_no	cgpa_10	cgpa_12	course_id	teacher_id	admission_status	max_cgpa	email	phone...	addre
1	101	Aarav Kapoor	123456789012	92	88	CS101	1	yes	100	NULL	NULL	NULL
2	102	Ishaan Malhotra	223456789012	81	79	ME102	2	yes	100	NULL	NULL	NULL
3	103	Riya Sharma	323456789012	82	85	EC103	3	yes	100	NULL	NULL	NULL
4	104	Ananya Gupta	423456789012	75	81	BT104	4	yes	100	NULL	NULL	NULL
5	105	Kabir Khanna	523456789012	70	75	CE105	5	yes	100	NULL	NULL	NULL
6	106	Devansh Nair	623456789012	90	93	IT106	6	yes	100	NULL	NULL	NULL
7	107	Aryan Sinha	723456789012	NULL	NULL	EE107	7	no	100	NULL	NULL	NULL
8	108	Tara Menon	823456789012	78	80	CH108	8	yes	100	NULL	NULL	NULL
9	109	Neel Patel	923456789012	NULL	NULL	MA109	9	no	100	NULL	NULL	NULL
10	110	Sanya Kapoor	103456789012	87	91	PH110	10	yes	100	NULL	NULL	NULL

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (69) | university | 00:00:00 | 10 rows

```
-- 2. SELECT specific columns: name and aadhaar number
SELECT name, aadhaar_no FROM students;
```

	name	aadhaar_no
1	Aarav Kapoor	123456789012
2	Ishaan Malhotra	223456789012
3	Riya Sharma	323456789012
4	Ananya Gupta	423456789012
5	Kabir Khanna	523456789012
6	Devansh Nair	623456789012
7	Aryan Sinha	723456789012
8	Tara Menon	823456789012
9	Neel Patel	923456789012
10	Sanya Kapoor	103456789012

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (69) | university | 00:00:00 | 10 rows

```
-- 3. SELECT DISTINCT course_id from students
SELECT DISTINCT course_id FROM students;
```

	course_id
1	BT104
2	CE105
3	CH108
4	CS101
5	EC103
6	EE107
7	IT106
8	MA109
9	ME102
10	PH110

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (69) | university | 00:00:00 | 10 rows

```sql
-- 4. LITERAL SELECT (no table)
SELECT 'Welcome to University DB' AS greeting;
```

	greeting
1	Welcome to University DB

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (69) | university | 00:00:00 | 1 rows

```sql
-- 5. SELECT rows where admission_status is 'yes'
SELECT * FROM students WHERE admission_status = 'yes';
```

	roll_no	name	aadhaar_no	cgpa_10	cgpa_12	course_id	teacher_id	admission_status	max_cgpa	email	phone_number
1	101	Aarav Kapoor	123456789012	92	88	CS101	1	yes	100	NULL	NULL
2	102	Ishaan Malhotra	223456789012	81	79	ME102	2	yes	100	NULL	NULL
3	103	Riya Sharma	323456789012	82	85	EC103	3	yes	100	NULL	NULL
4	104	Ananya Gupta	423456789012	75	81	BT104	4	yes	100	NULL	NULL
5	105	Kabir Khanna	523456789012	70	75	CE105	5	yes	100	NULL	NULL
6	106	Devansh Nair	623456789012	90	93	IT106	6	yes	100	NULL	NULL
7	108	Tara Menon	823456789012	78	80	CH108	8	yes	100	NULL	NULL
8	110	Sanya Kapoor	103456789012	87	91	PH110	10	yes	100	NULL	NULL

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (69) | university | 00:00:00 | 8 rows

```sql
-- 6. SELECT rows ordered by cgpa_10 in descending order
SELECT * FROM students ORDER BY cgpa_10 DESC;
```

	roll_no	name	aadhaar_no	cgpa_10	cgpa_12	course_id	teacher_id	admission_status	max_cgpa	email	phone_number
1	101	Aarav Kapoor	123456789012	92	88	CS101	1	yes	100	NULL	NULL
2	106	Devansh Nair	623456789012	90	93	IT106	6	yes	100	NULL	NULL
3	110	Sanya Kapoor	103456789012	87	91	PH110	10	yes	100	NULL	NULL
4	103	Riya Sharma	323456789012	82	85	EC103	3	yes	100	NULL	NULL
5	102	Ishaan Malhotra	223456789012	81	79	ME102	2	yes	100	NULL	NULL
6	108	Tara Menon	823456789012	78	80	CH108	8	yes	100	NULL	NULL
7	104	Ananya Gupta	423456789012	75	81	BT104	4	yes	100	NULL	NULL
8	105	Kabir Khanna	523456789012	70	75	CE105	5	yes	100	NULL	NULL
9	109	Neel Patel	923456789012	NULL	NULL	MA109	9	no	100	NULL	NULL
10	107	Aryan Sinha	723456789012	NULL	NULL	EE107	7	no	100	NULL	NULL

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (69) | university | 00:00:00 | 10 rows

```sql
-- 7. SELECT with CASE to classify performance
SELECT name,
cgpa_10,
CASE
WHEN cgpa_10 >= 9 THEN 'Excellent'
WHEN cgpa_10 >= 7 THEN 'Good'
ELSE 'Average'
END AS performance
FROM students;
```

	name	cgpa_10	performance
1	Aarav Kapoor	92	Excellent
2	Ishaan Malhotra	81	Excellent
3	Riya Sharma	82	Excellent
4	Ananya Gupta	75	Excellent
5	Kabir Khanna	70	Excellent
6	Devansh Nair	90	Excellent
7	Aryan Sinha	NULL	Average
8	Tara Menon	78	Excellent
9	Neel Patel	NULL	Average
10	Sanya Kapoor	87	Excellent

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (69) | university | 00:00:00 | 10 rows

```sql
-- 8. SELECT rows with teacher_id = 4 and order by name
ascending
SELECT * FROM students WHERE teacher_id = 4 ORDER BY name ASC;
```

	roll_no	name	aadhaar_no	cgpa_10	cgpa_12	course_id	teacher_id	admission_status	max_cgpa	email	phone_number	a
1	104	Ananya Gupta	423456789012	75	81	BT104	4	yes	100	NULL	NULL	

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (69) | university | 00:00:00 | 1 rows

```sql
-- 9. SELECT specific columns with aliases
```

```sql
SELECT name AS student_name, cgpa_12 AS twelfth_cgpa FROM
students;
```

	student_name	twelfth_cgpa
1	Aarav Kapoor	88
2	Ishaan Malhotra	79
3	Riya Sharma	85
4	Ananya Gupta	81
5	Kabir Khanna	75
6	Devansh Nair	93
7	Aryan Sinha	NULL
8	Tara Menon	80
9	Neel Patel	NULL
10	Sanya Kapoor	91

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (69) | university | 00:00:00 | 10 rows

```sql
-- 10. SELECT where cgpa_10 is between 70 and 90
SELECT * FROM students WHERE cgpa_10 BETWEEN 70 AND 90;
```

	roll_no	name	aadhaar_no	cgpa_10	cgpa_12	course_id	teacher_id	admission_status	max_cgpa	email	phone_number
1	101	Aarav Kapoor	123456789012	92	88	CS101	1	yes	100	NULL	NULL
2	102	Ishaan Malhotra	223456789012	81	79	ME102	2	yes	100	NULL	NULL
3	103	Riya Sharma	323456789012	82	85	EC103	3	yes	100	NULL	NULL
4	104	Ananya Gupta	423456789012	75	81	BT104	4	yes	100	NULL	NULL
5	105	Kabir Khanna	523456789012	70	75	CE105	5	yes	100	NULL	NULL
6	108	Tara Menon	823456789012	78	80	CH108	8	yes	100	NULL	NULL

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (69) | university | 00:00:00 | 6 rows

```sql
-- 11. SELECT students where cgpa_12 is NULL
SELECT * FROM students WHERE cgpa_12 IS NULL;
```

	roll_no	name	aadhaar_no	cgpa_10	cgpa_12	course_id	teacher_id	admission_status	max_cgpa	email	phone_number	add
1	107	Aryan Sinha	723456789012	NULL	NULL	EE107	7	no	100	NULL	NULL	NU
2	109	Neel Patel	923456789012	NULL	NULL	MA109	9	no	100	NULL	NULL	NU

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (69) | university | 00:00:00 | 2 rows

```sql
-- 12. SELECT students in CS101 or ME102 course
SELECT * FROM students WHERE course_id IN ('CS101', 'ME102');
```

	roll_no	name	aadhaar_no	cgpa_10	cgpa_12	course_id	teacher_id	admission_status	max_cgpa	email	phone_number
1	101	Aarav Kapoor	123456789012	92	88	CS101	1	yes	100	NULL	NULL
2	102	Ishaan Malhotra	223456789012	81	79	ME102	2	yes	100	NULL	NULL

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (69) | university | 00:00:00 | 2 rows

```sql
-- 13. SELECT students with names starting with 'A'
SELECT * FROM students WHERE name LIKE 'A%';
```

	roll_no	name	aadhaar_no	cgpa_10	cgpa_12	course_id	teacher_id	admission_status	max_cgpa	email	phone_number	a
1	101	Aarav Kapoor	123456789012	92	88	CS101	1	yes	100	NULL	NULL	
2	104	Ananya Gupta	423456789012	75	81	BT104	4	yes	100	NULL	NULL	
3	107	Aryan Sinha	723456789012	NULL	NULL	EE107	7	no	100	NULL	NULL	

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (69) | university | 00:00:00 | 3 rows

```sql
-- 14. SELECT all data from teachers table
SELECT * FROM teachers;
```

	teacher_id	course_id	name
1	1	CS101	Rajesh Sharma
2	2	ME102	Anil Kumar
3	3	EC103	Priya Mehta
4	4	BT104	Amitabh Joshi
5	5	CE105	Neha Reddy
6	6	IT106	Arun Mishra
7	7	EE107	Pooja Patel
8	8	CH108	Ravi Verma
9	9	MA109	Sanjay Gupta
10	10	PH110	Kavita Iyer

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (69) | university | 00:00:00 | 10 rows

```sql
-- 15. SELECT with CASE to determine eligibility based on
admission_status
SELECT name,
admission_status,
CASE
WHEN admission_status = 'yes' THEN 'Eligible'
ELSE 'Not Eligible'
END AS eligibility
FROM students
ORDER BY name;
```

	name	admission_status	eligibility
1	Aarav Kapoor	yes	Eligible
2	Ananya Gupta	yes	Eligible
3	Aryan Sinha	no	Not Eligible
4	Devansh Nair	yes	Eligible
5	Ishaan Malhotra	yes	Eligible
6	Kabir Khanna	yes	Eligible
7	Neel Patel	no	Not Eligible
8	Riya Sharma	yes	Eligible
9	Sanya Kapoor	yes	Eligible
10	Tara Menon	yes	Eligible

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (69) | university | 00:00:00 | 10 rows

5. Perform at least 15 queries using keywords SELECT, FROM, OFFSET and FETCH, LIMIT and pattern matching.

```
/*
Step: Perform at least 15 queries using OFFSET & FETCH,
pattern matching (LIKE), and LIMIT-equivalent (TOP).
Database: university
*/

USE university;

-- 1. Get first 5 students (LIMIT equivalent in SQL Server)
SELECT TOP 5 * FROM students;
```

	roll_no	name	aadhaar_no	cgpa_10	cgpa_12	course_id	teacher_id	admission_status	max_cgpa
1	101	Aarav Kapoor	123456789012	9.2	8.8	CS101	1	yes	10
2	102	Ishaan Malhotra	223456789012	8.1	7.9	ME102	2	yes	10
3	103	Riya Sharma	323456789012	NULL	NULL	EC103	3	no	10
4	104	Ananya Gupta	423456789012	7.5	8.1	BT104	4	yes	10
5	105	Kabir Khanna	523456789012	NULL	NULL	CE105	5	no	10

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (51) | university | 00:00:00 | 5 rows

```
-- 2. Get next 5 students after skipping first 5 (OFFSET +
FETCH)
SELECT * FROM students
ORDER BY roll_no
OFFSET 5 ROWS
FETCH NEXT 5 ROWS ONLY;
```

	roll_no	name	aadhaar_no	cgpa_10	cgpa_12	course_id	teacher_id	admission_status	max_cgpa
1	106	Devansh Nair	623456789012	9	9.3	IT106	6	yes	10
2	107	Aryan Sinha	723456789012	NULL	NULL	EE107	7	no	10
3	108	Tara Menon	823456789012	7.8	8	CH108	8	yes	10
4	109	Neel Patel	923456789012	NULL	NULL	MA109	9	no	10
5	110	Sanya Kapoor	103456789012	8.7	9.1	PH110	10	yes	10

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (51) | university | 00:00:00 | 5 rows

```
-- 3. Get students where name starts with 'A' (pattern
matching)
SELECT * FROM students WHERE name LIKE 'A%';
```

	roll_no	name	aadhaar_no	cgpa_10	cgpa_12	course_id	teacher_id	admission_status	max_cgpa
1	101	Aarav Kapoor	123456789012	9.2	8.8	CS101	1	yes	10
2	104	Ananya Gupta	423456789012	7.5	8.1	BT104	4	yes	10
3	107	Aryan Sinha	723456789012	NULL	NULL	EE107	7	no	10

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (51) | university | 00:00:00 | 3 rows

```
-- 4. Get students where name ends with 'a'
```

```sql
SELECT * FROM students WHERE name LIKE '%a';
```

	roll_no	name	aadhaar_no	cgpa_10	cgpa_12	course_id	teacher_id	admission_status	max_cgpa
1	102	Ishaan Malhotra	223456789012	8.1	7.9	ME102	2	yes	10
2	103	Riya Sharma	323456789012	NULL	NULL	EC103	3	no	10
3	104	Ananya Gupta	423456789012	7.5	8.1	BT104	4	yes	10
4	105	Kabir Khanna	523456789012	NULL	NULL	CE105	5	no	10
5	107	Aryan Sinha	723456789012	NULL	NULL	EE107	7	no	10

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (51) | university | 00:00:00 | 5 rows

```sql
-- 5. Get students whose name contains 'an'
SELECT * FROM students WHERE name LIKE '%an%';
```

	roll_no	name	aadhaar_no	cgpa_10	cgpa_12	course_id	teacher_id	admission_status	max_cgpa
1	102	Ishaan Malhotra	223456789012	8.1	7.9	ME102	2	yes	10
2	104	Ananya Gupta	423456789012	7.5	8.1	BT104	4	yes	10
3	105	Kabir Khanna	523456789012	NULL	NULL	CE105	5	no	10
4	106	Devansh Nair	623456789012	9	9.3	IT106	6	yes	10
5	107	Aryan Sinha	723456789012	NULL	NULL	EE107	7	no	10
6	110	Sanya Kapoor	103456789012	8.7	9.1	PH110	10	yes	10

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (51) | university | 00:00:00 | 6 rows

```sql
-- 6. Get students whose name has second letter 'a'
SELECT * FROM students WHERE name LIKE '_a%';
```

	roll_no	name	aadhaar_no	cgpa_10	cgpa_12	course_id	teacher_id	admission_status	max_cgpa
1	101	Aarav Kapoor	123456789012	9.2	8.8	CS101	1	yes	10
2	105	Kabir Khanna	523456789012	NULL	NULL	CE105	5	no	10
3	108	Tara Menon	823456789012	7.8	8	CH108	8	yes	10
4	110	Sanya Kapoor	103456789012	8.7	9.1	PH110	10	yes	10

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (51) | university | 00:00:00 | 4 rows

```sql
-- 7. Get teachers where teacher_name starts with 'P'
SELECT * FROM teachers WHERE teacher_name LIKE 'P%';
```

	teacher_id	course_id	teacher_name
1	3	EC103	Priya Mehta
2	7	EE107	Pooja Patel

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (51) | university | 00:00:00 | 2 rows

```sql
-- 8. Get teachers whose name ends with 'a'
SELECT * FROM teachers WHERE teacher_name LIKE '%a';
```

	teacher_id	course_id	teacher_name
1	1	CS101	Rajesh Sharma
2	3	EC103	Priya Mehta
3	6	IT106	Arun Mishra
4	8	CH108	Ravi Verma
5	9	MA109	Sanjay Gupta

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (51) | university | 00:00:00 | 5 rows

```sql
-- 9. Get students with non-null CGPA_10 and order them with
OFFSET-FETCH
```

```sql
SELECT * FROM students
WHERE cgpa_10 IS NOT NULL
ORDER BY cgpa_10 DESC
OFFSET 2 ROWS
FETCH NEXT 3 ROWS ONLY;
```

	roll_no	name	aadhaar_no	cgpa_10	cgpa_12	course_id	teacher_id	admission_status	max_cgpa
1	110	Sanya Kapoor	103456789012	8.7	9.1	PH110	10	yes	10
2	102	Ishaan Malhotra	223456789012	8.1	7.9	ME102	2	yes	10
3	108	Tara Menon	823456789012	7.8	8	CH108	8	yes	10

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (51) | university | 00:00:00 | 3 rows

```sql
-- 10. Get students with names containing 'ar'
SELECT * FROM students WHERE name LIKE '%ar%';
```

	roll_no	name	aadhaar_no	cgpa_10	cgpa_12	course_id	teacher_id	admission_status	max_cgpa
1	101	Aarav Kapoor	123456789012	9.2	8.8	CS101	1	yes	10
2	103	Riya Sharma	323456789012	NULL	NULL	EC103	3	no	10
3	107	Aryan Sinha	723456789012	NULL	NULL	EE107	7	no	10
4	108	Tara Menon	823456789012	7.8	8	CH108	8	yes	10

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (51) | university | 00:00:00 | 4 rows

```sql
-- 11. Use OFFSET to skip top 3 scorers and fetch next 2
SELECT * FROM students
WHERE cgpa_10 IS NOT NULL
ORDER BY cgpa_10 DESC
OFFSET 3 ROWS
FETCH NEXT 2 ROWS ONLY;
```

	roll_no	name	aadhaar_no	cgpa_10	cgpa_12	course_id	teacher_id	admission_status	max_cgpa
1	102	Ishaan Malhotra	223456789012	8.1	7.9	ME102	2	yes	10
2	108	Tara Menon	823456789012	7.8	8	CH108	8	yes	10

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (51) | university | 00:00:00 | 2 rows

```sql
-- 12. Get top 1 student with highest cgpa_12
SELECT TOP 1 * FROM students
WHERE cgpa_12 IS NOT NULL
ORDER BY cgpa_12 DESC;
```

	roll_no	name	aadhaar_no	cgpa_10	cgpa_12	course_id	teacher_id	admission_status	max_cgpa
1	106	Devansh Nair	623456789012	9	9.3	IT106	6	yes	10

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (51) | university | 00:00:00 | 1 rows

```sql
-- 13. Get names of students whose Aadhaar contains '56'
SELECT name, aadhaar_no FROM students WHERE aadhaar_no LIKE
'%56%';
```

	name	aadhaar_no
1	Aarav Kapoor	123456789012
2	Ishaan Malhotra	223456789012
3	Riya Sharma	323456789012
4	Ananya Gupta	423456789012
5	Kabir Khanna	523456789012
6	Devansh Nair	623456789012
7	Aryan Sinha	723456789012
8	Tara Menon	823456789012
9	Neel Patel	923456789012
10	Sanya Kapoor	103456789012

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (51) | university | 00:00:00 | 10 rows

```sql
-- 14. Get teachers with course_id ending in '1'
SELECT * FROM teachers WHERE course_id LIKE '%1';
```

	teacher_id	course_id	teacher_name
1	1	CS101	Rajesh Sharma

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (51) | university | 00:00:00 | 1 rows

```sql
-- 15. Get 3 students after skipping first 2 alphabetically
SELECT * FROM students
ORDER BY name ASC
OFFSET 2 ROWS
FETCH NEXT 3 ROWS ONLY;
```

	roll_no	name	aadhaar_no	cgpa_10	cgpa_12	course_id	teacher_id	admission_status	max_cgpa
1	107	Aryan Sinha	723456789012	NULL	NULL	EE107	7	no	10
2	106	Devansh Nair	623456789012	9	9.3	IT106	6	yes	10
3	102	Ishaan Malhotra	223456789012	8.1	7.9	ME102	2	yes	10

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (51) | university | 00:00:00 | 3 rows

6. Perform at least 15 queries using keywords, GROUP BY, HAVING and ORDER BY.

```sql
-- 1. Total students per teacher
SELECT teacher_id, COUNT(*) AS total_students
FROM students
GROUP BY teacher_id
ORDER BY total_students DESC;
```

	teacher_id	total_students
1	1	1
2	2	1
3	3	1
4	4	1
5	5	1
6	6	1
7	7	1
8	8	1
9	9	1
10	10	1

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 10 rows

```sql
-- 2. Average CGPA_10 per teacher
SELECT teacher_id, AVG(cgpa_10) AS avg_cgpa_10
FROM students
WHERE cgpa_10 IS NOT NULL
GROUP BY teacher_id
ORDER BY avg_cgpa_10 DESC;
```

	teacher_id	avg_cgpa_10
1	1	9.2
2	6	9
3	10	8.7
4	2	8.1
5	8	7.8
6	4	7.5

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 6 rows

```sql
-- 3. Students per course
SELECT course_id, COUNT(*) AS total_students
FROM students
GROUP BY course_id
ORDER BY course_id;
```

	course_id	total_students
1	BT104	1
2	CE105	1
3	CH108	1
4	CS101	1
5	EC103	1
6	EE107	1
7	IT106	1
8	MA109	1
9	ME102	1
10	PH110	1

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 10 rows

```sql
-- 4. Number of students with admission_status = 'yes' per teacher
SELECT teacher_id, COUNT(*) AS admitted_students
FROM students
WHERE admission_status = 'yes'
GROUP BY teacher_id
ORDER BY teacher_id;
```

	teacher_id	admitted_students
1	1	1
2	2	1
3	4	1
4	6	1
5	8	1
6	10	1

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 6 rows

```sql
-- 5. Maximum CGPA_12 per teacher
SELECT teacher_id, MAX(cgpa_12) AS max_cgpa_12
FROM students
WHERE cgpa_12 IS NOT NULL
GROUP BY teacher_id;
```

	teacher_id	max_cgpa_12
1	1	8.8
2	2	7.9
3	4	8.1
4	6	9.3
5	8	8
6	10	9.1

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 6 rows

```sql
-- 6. All courses with number of students
SELECT course_id, COUNT(*) AS total_students
FROM students
GROUP BY course_id;
```

	course_id	total_students
1	BT104	1
2	CE105	1
3	CH108	1
4	CS101	1
5	EC103	1
6	EE107	1
7	IT106	1
8	MA109	1
9	ME102	1
10	PH110	1

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 10 rows

```sql
-- 7. Number of students having CGPA_10 >= 8 grouped by course
SELECT course_id, COUNT(*) AS high_cgpa_students
FROM students
WHERE cgpa_10 >= 8
GROUP BY course_id;
```

	course_id	high_cgpa_students
1	CS101	1
2	IT106	1
3	ME102	1
4	PH110	1

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 4 rows

```sql
-- 8. Group by admission_status and show average CGPA_12
SELECT admission_status, AVG(cgpa_12) AS avg_cgpa_12
FROM students
WHERE cgpa_12 IS NOT NULL
GROUP BY admission_status;
```

	admission_status	avg_cgpa_12
1	yes	8.53333333333334

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 1 rows

```sql
-- 9. Total students grouped by Aadhaar number pattern (first
digit)
SELECT LEFT(aadhaar_no, 1) AS aadhaar_prefix, COUNT(*) AS
count
FROM students
GROUP BY LEFT(aadhaar_no, 1)
ORDER BY aadhaar_prefix;
```

	aadhaar_prefix	count
1	1	2
2	2	1
3	3	1
4	4	1
5	5	1
6	6	1
7	7	1
8	8	1
9	9	1

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 9 rows

```sql
-- 10. Student count per teacher
SELECT teacher_id, COUNT(*) AS students
FROM students
GROUP BY teacher_id;
```

	teacher_id	students
1	1	1
2	2	1
3	3	1
4	4	1
5	5	1
6	6	1
7	7	1
8	8	1
9	9	1
10	10	1

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 10 rows

```sql
-- 11. Group students by max_cgpa and show count
SELECT max_cgpa, COUNT(*) AS count
FROM students
GROUP BY max_cgpa;
```

	max_cgpa	count
1	10	10

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 1 rows

```sql
-- 12. Students with CGPA_12 above 8 per teacher
SELECT teacher_id, COUNT(*) AS top_students
FROM students
WHERE cgpa_12 > 8
GROUP BY teacher_id;
```

	teacher_id	top_students
1	1	1
2	4	1
3	6	1
4	10	1

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 4 rows

```sql
-- 13. Group by course_id and show minimum cgpa_10
SELECT course_id, MIN(cgpa_10) AS min_cgpa_10
FROM students
```

```
WHERE cgpa_10 IS NOT NULL
GROUP BY course_id;
```

	course_id	min_cgpa_10
1	BT104	7.5
2	CH108	7.8
3	CS101	9.2
4	IT106	9
5	ME102	8.1
6	PH110	8.7

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 6 rows

```
-- 14. Group by course_id and admission_status together
SELECT course_id, admission_status, COUNT(*) AS count
FROM students
GROUP BY course_id, admission_status
ORDER BY course_id;
```

	course_id	admission_status	count
1	BT104	yes	1
2	CE105	no	1
3	CH108	yes	1
4	CS101	yes	1
5	EC103	no	1
6	EE107	no	1
7	IT106	yes	1
8	MA109	no	1
9	ME102	yes	1
10	PH110	yes	1

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 10 rows

```
-- 15. Average CGPA_10 per teacher with students
SELECT teacher_id, AVG(cgpa_10) AS avg_cgpa
FROM students
WHERE cgpa_10 IS NOT NULL
GROUP BY teacher_id;
```

	teacher_id	avg_cgpa
1	1	9.2
2	2	8.1
3	4	7.5
4	6	9
5	8	7.8
6	10	8.7

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 6 rows

7. Perform at least 20 queries using combination of all the above keywords/concepts.

```
USE university

-- 1. List all distinct course IDs
SELECT DISTINCT course_id FROM students;
```

	course_id
1	BT104
2	CE105
3	CH108
4	CS101
5	EC103
6	EE107
7	IT106
8	MA109
9	ME102
10	PH110

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 10 rows

```
-- 2. Display student name and status label using CASE
SELECT name,
CASE
WHEN admission_status = 'yes' THEN 'Admitted'
ELSE 'Not Admitted'
END AS status
FROM students;
```

	name	status
1	Aarav Kapoor	Admitted
2	Ishaan Malhotra	Admitted
3	Riya Sharma	Not Admitted
4	Ananya Gupta	Admitted
5	Kabir Khanna	Not Admitted
6	Devansh Nair	Admitted
7	Aryan Sinha	Not Admitted
8	Tara Menon	Admitted
9	Neel Patel	Not Admitted
10	Sanya Kapoor	Admitted

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 10 rows

```
-- 3. List students with CGPA_10 >= 8, ordered by CGPA_10
descending
SELECT name, cgpa_10
FROM students
WHERE cgpa_10 >= 8
ORDER BY cgpa_10 DESC;
```

	name	cgpa_10
1	Aarav Kapoor	9.2
2	Devansh Nair	9
3	Sanya Kapoor	8.7
4	Ishaan Malhotra	8.1

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 4 rows

```sql
-- 4. Count of students per admission_status
SELECT admission_status, COUNT(*) AS count
FROM students
GROUP BY admission_status;
```

	admission_status	count
1	no	4
2	yes	6

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 2 rows

```sql
-- 5. Average CGPA_12 by admission_status, only where avg > 8
SELECT admission_status, AVG(cgpa_12) AS avg_12
FROM students
WHERE cgpa_12 IS NOT NULL
GROUP BY admission_status
HAVING AVG(cgpa_12) > 8;
```

	admission_status	avg_12
1	yes	8.53333333333334

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 1 rows

```sql
-- 6. List of teachers with students admitted
SELECT DISTINCT t.teacher_name
FROM teachers t
JOIN students s ON t.teacher_id = s.teacher_id
WHERE s.admission_status = 'yes';
```

	teacher_name
1	Amitabh Joshi
2	Anil Kumar
3	Arun Mishra
4	Kavita Iyer
5	Rajesh Shar...
6	Ravi Verma

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 6 rows

```sql
-- 7. Show first 5 students alphabetically
SELECT * FROM students
ORDER BY name
OFFSET 0 ROWS FETCH NEXT 5 ROWS ONLY;
```

	roll_no	name	aadhaar_no	cgpa_10	cgpa_12	course_id	teacher_id	admission_status	max_cgpa
1	101	Aarav Kapoor	123456789012	9.2	8.8	CS101	1	yes	10
2	104	Ananya Gupta	423456789012	7.5	8.1	BT104	4	yes	10
3	107	Aryan Sinha	723456789012	NULL	NULL	EE107	7	no	10
4	106	Devansh Nair	623456789012	9	9.3	IT106	6	yes	10
5	102	Ishaan Malhotra	223456789012	8.1	7.9	ME102	2	yes	10

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 5 rows

```sql
-- 8. Show students whose Aadhaar ends in '12'
SELECT * FROM students
WHERE aadhaar_no LIKE '%12';
```

	roll_no	name	aadhaar_no	cgpa_10	cgpa_12	course_id	teacher_id	admission_status	max_cgpa
1	101	Aarav Kapoor	123456789012	9.2	8.8	CS101	1	yes	10
2	102	Ishaan Malhotra	223456789012	8.1	7.9	ME102	2	yes	10
3	103	Riya Sharma	323456789012	NULL	NULL	EC103	3	no	10
4	104	Ananya Gupta	423456789012	7.5	8.1	BT104	4	yes	10
5	105	Kabir Khanna	523456789012	NULL	NULL	CE105	5	no	10
6	106	Devansh Nair	623456789012	9	9.3	IT106	6	yes	10
7	107	Aryan Sinha	723456789012	NULL	NULL	EE107	7	no	10
8	108	Tara Menon	823456789012	7.8	8	CH108	8	yes	10
9	109	Neel Patel	923456789012	NULL	NULL	MA109	9	no	10
10	110	Sanya Kapoor	103456789012	8.7	9.1	PH110	10	yes	10

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 10 rows

```sql
-- 9. List of students with NULL in cgpa_10
SELECT name FROM students
WHERE cgpa_10 IS NULL;
```

	name
1	Riya Sharma
2	Kabir Khanna
3	Aryan Sinha
4	Neel Patel

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 4 rows

```sql
-- 10. Count students per teacher, but only show those with more than 0
SELECT teacher_id, COUNT(*) AS student_count
FROM students
GROUP BY teacher_id
HAVING COUNT(*) > 0;
```

	teacher_id	student_count
1	1	1
2	2	1
3	3	1
4	4	1
5	5	1
6	6	1
7	7	1
8	8	1
9	9	1
10	10	1

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 10 rows

```sql
-- 11. Students whose names start with 'A'
SELECT * FROM students
WHERE name LIKE 'A%';
```

	roll_no	name	aadhaar_no	cgpa_10	cgpa_12	course_id	teacher_id	admission_status	max_cgpa
1	101	Aarav Kapoor	123456789012	9.2	8.8	CS101	1	yes	10
2	104	Ananya Gupta	423456789012	7.5	8.1	BT104	4	yes	10
3	107	Aryan Sinha	723456789012	NULL	NULL	EE107	7	no	10

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 3 rows

```sql
-- 12. Top 3 CGPA_12 holders
SELECT name, cgpa_12
FROM students
```

```sql
WHERE cgpa_12 IS NOT NULL
ORDER BY cgpa_12 DESC
OFFSET 0 ROWS FETCH NEXT 3 ROWS ONLY;
```

	name	cgpa_12
1	Devansh Nair	9.3
2	Sanya Kapoor	9.1
3	Aarav Kapoor	8.8

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 3 rows

```sql
-- 13. List courses and how many students have CGPA_10 >= 8
SELECT course_id, COUNT(*) AS high_achievers
FROM students
WHERE cgpa_10 >= 8
GROUP BY course_id;
```

	course_id	high_achievers
1	CS101	1
2	IT106	1
3	ME102	1
4	PH110	1

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 4 rows

```sql
-- 14. Count how many courses start with 'C'
SELECT COUNT(DISTINCT course_id) AS count
FROM students
WHERE course_id LIKE 'C%';
```

	count
1	3

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 1 rows

```sql
-- 15. Show teacher name and number of admitted students
SELECT t.teacher_name, COUNT(*) AS admitted
FROM teachers t
JOIN students s ON t.teacher_id = s.teacher_id
WHERE s.admission_status = 'yes'
GROUP BY t.teacher_name;
```

	teacher_name	admitted
1	Amitabh Joshi	1
2	Anil Kumar	1
3	Arun Mishra	1
4	Kavita Iyer	1
5	Rajesh Shar...	1
6	Ravi Verma	1

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 6 rows

```sql
-- 16. List students ordered by CGPA_10 with NULLs last
SELECT name, cgpa_10
FROM students
ORDER BY
CASE WHEN cgpa_10 IS NULL THEN 1 ELSE 0 END,
```

```
cgpa_10 DESC;
```

	name	cgpa_10
1	Aarav Kapoor	9.2
2	Devansh Nair	9
3	Sanya Kapoor	8.7
4	Ishaan Malhotra	8.1
5	Tara Menon	7.8
6	Ananya Gupta	7.5
7	Kabir Khanna	NULL
8	Riya Sharma	NULL
9	Neel Patel	NULL
10	Aryan Sinha	NULL

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 10 rows

```
-- 17. Students grouped by course and admission status
SELECT course_id, admission_status, COUNT(*) AS count
FROM students
GROUP BY course_id, admission_status
ORDER BY course_id;
```

	course_id	admission_status	count
1	BT104	yes	1
2	CE105	no	1
3	CH108	yes	1
4	CS101	yes	1
5	EC103	no	1
6	EE107	no	1
7	IT106	yes	1
8	MA109	no	1
9	ME102	yes	1
10	PH110	yes	1

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 10 rows

```
-- 18. List student names with CGPA_10 > CGPA_12 (only if both
are not null)
SELECT name, cgpa_10, cgpa_12
FROM students
WHERE cgpa_10 IS NOT NULL AND cgpa_12 IS NOT NULL AND cgpa_10
> cgpa_12;
```

	name	cgpa_10	cgpa_12
1	Aarav Kapoor	9.2	8.8
2	Ishaan Malhotra	8.1	7.9

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 2 rows

```
-- 19. Show students with cgpa_10 greater than 7 and name
ending with 'a'.
SELECT * FROM students
WHERE cgpa_10>7 AND name LIKE '%a';
```

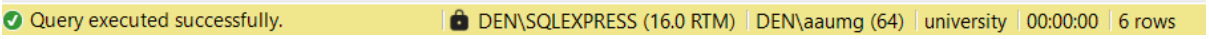
	roll_no	name	aadhaar_no	cgpa_10	cgpa_12	course_id	teacher_id	admission_status	max_cgpa
1	102	Ishaan Malhotra	223456789012	8.1	7.9	ME102	2	yes	10
2	104	Ananya Gupta	423456789012	7.5	8.1	BT104	4	yes	10

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (64) | university | 00:00:00 | 2 rows

```
-- 20. Show student names with roll_no between 103 and 108
```

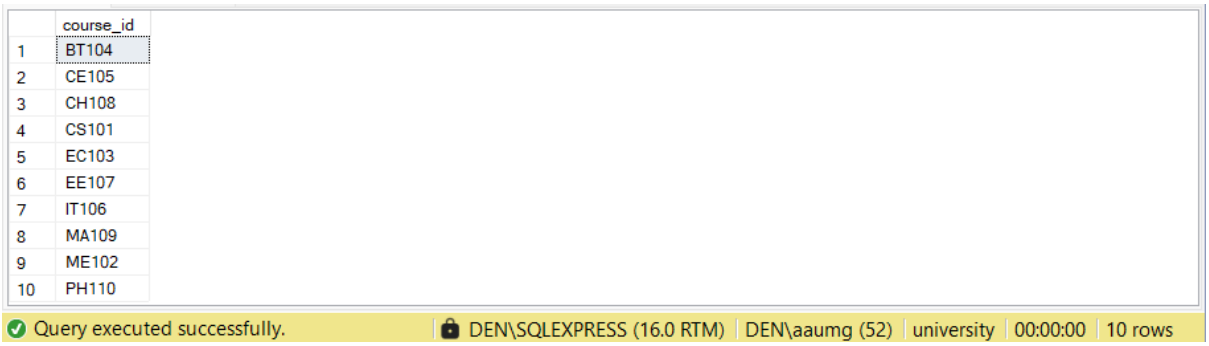
```sql
SELECT name, roll_no
FROM students
WHERE roll_no BETWEEN 103 AND 108
ORDER BY roll_no;
```

	name	roll_no
1	Riya Sharma	103
2	Ananya Gupta	104
3	Kabir Khanna	105
4	Devansh Nair	106
5	Aryan Sinha	107
6	Tara Menon	108



8. Perform at least 10 queries based on set operations.

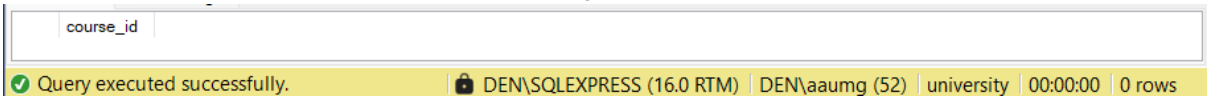
```
USE university

-- 1. Get course_ids present in both teachers and students
(INTERSECT)
SELECT course_id FROM students
INTERSECT
SELECT course_id FROM teachers;
```



	course_id
1	BT104
2	CE105
3	CH108
4	CS101
5	EC103
6	EE107
7	IT106
8	MA109
9	ME102
10	PH110

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (52) | university | 00:00:00 | 10 rows

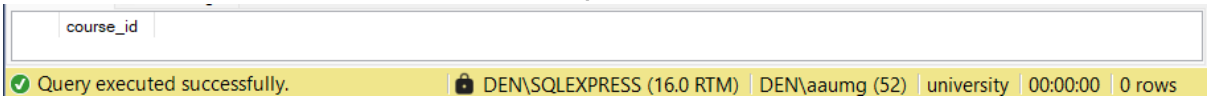
```
-- 2. Get course_ids that are in students but not in teachers
(EXCEPT)
SELECT course_id FROM students
EXCEPT
SELECT course_id FROM teachers;
```



course_id

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (52) | university | 00:00:00 | 0 rows

```
-- 3. Get course_ids that are in teachers but not in students
(EXCEPT)
SELECT course_id FROM teachers
EXCEPT
SELECT course_id FROM students;
```



course_id

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (52) | university | 00:00:00 | 0 rows

```
-- 4. List all course_ids from both tables (UNION)
SELECT course_id FROM students
UNION
SELECT course_id FROM teachers;
```

	course_id
1	BT104
2	CE105
3	CH108
4	CS101
5	EC103
6	EE107
7	IT106
8	MA109
9	ME102
10	PH110

Query executed successfully. DEN\SQLEXPRESS (16.0 RTM) DEN\aaumg (52) university 00:00:00 10 rows

```sql
-- 5. List all course_ids from both tables (duplicates allowed - UNION ALL)
SELECT course_id FROM students
UNION ALL
SELECT course_id FROM teachers;
```

	course_id
1	CS101
2	ME102
3	EC103
4	BT104
5	CE105
6	IT106
7	EE107
8	CH108
9	MA109
10	PH110

Query executed successfully. DEN\SQLEXPRESS (16.0 RTM) DEN\aaumg (52) university 00:00:00 20 rows

```sql
-- 6. List all teacher_ids who also appear in students table (INTERSECT)
SELECT teacher_id FROM teachers
INTERSECT
SELECT teacher_id FROM students;
```

	teacher_id
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

Query executed successfully. DEN\SQLEXPRESS (16.0 RTM) DEN\aaumg (52) university 00:00:00 10 rows

```sql
-- 7. List names of students and teacher_names together (UNION ALL)
SELECT name AS person FROM students
UNION ALL
SELECT teacher_name FROM teachers;
```

	person
1	Aarav Kapoor
2	Ishaan Malhotra
3	Riya Sharma
4	Ananya Gupta
5	Kabir Khanna
6	Devansh Nair
7	Aryan Sinha
8	Tara Menon
9	Neel Patel
10	Sanya Kapoor
11	Rajesh Sharma
12	Anil Kumar
13	Priya Mehta
14	Amitabh Joshi
15	Neha Reddy
16	Arun Mishra
17	Pooja Patel
18	Ravi Verma
19	Sanjay Gupta
20	Kavita Iyer

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (52) | university | 00:00:00 | 20 rows

```sql
-- 8. Roll numbers of students with cgpa_10 and cgpa_12 not null (INTERSECT of two filtered sets)
SELECT roll_no FROM students WHERE cgpa_10 IS NOT NULL
INTERSECT
SELECT roll_no FROM students WHERE cgpa_12 IS NOT NULL;
```

	roll_no
1	101
2	102
3	104
4	106
5	108
6	110

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (52) | university | 00:00:00 | 6 rows

```sql
-- 9. Students who are admitted (yes) but have no CGPA_10 set (set difference)
SELECT roll_no FROM students WHERE admission_status = 'yes'
EXCEPT
SELECT roll_no FROM students WHERE cgpa_10 IS NOT NULL;
```

	roll_no

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (52) | university | 00:00:00 | 0 rows

```sql
-- 10. Combine teacher and student IDs into one column (UNION)
SELECT teacher_id AS id FROM teachers
UNION
SELECT teacher_id FROM students;
```

	id
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (52) | university | 00:00:00 | 10 rows

9. Perform at least 15 queries using joins.

```sql
-- 1. INNER JOIN: Get student name, course ID, and corresponding teacher name
SELECT s.name, s.course_id, t.teacher_name
FROM students s
INNER JOIN teachers t ON s.teacher_id = t.teacher_id;
```

	name	course_id	teacher_name
1	Aarav Kapoor	CS101	Rajesh Sharma
2	Ishaan Malhotra	ME102	Anil Kumar
3	Riya Sharma	EC103	Priya Mehta
4	Ananya Gupta	BT104	Amitabh Joshi
5	Kabir Khanna	CE105	Neha Reddy
6	Devansh Nair	IT106	Arun Mishra
7	Aryan Sinha	EE107	Pooja Patel
8	Tara Menon	CH108	Ravi Verma
9	Neel Patel	MA109	Sanjay Gupta
10	Sanya Kapoor	PH110	Kavita Iyer

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (52) | university | 00:00:00 | 10 rows

```sql
-- 2. LEFT JOIN: List all students and their teacher names (if any)
SELECT s.name, t.teacher_name
FROM students s
LEFT JOIN teachers t ON s.teacher_id = t.teacher_id;
```

	name	teacher_name
1	Aarav Kapoor	Rajesh Sharma
2	Ishaan Malhotra	Anil Kumar
3	Riya Sharma	Priya Mehta
4	Ananya Gupta	Amitabh Joshi
5	Kabir Khanna	Neha Reddy
6	Devansh Nair	Arun Mishra
7	Aryan Sinha	Pooja Patel
8	Tara Menon	Ravi Verma
9	Neel Patel	Sanjay Gupta
10	Sanya Kapoor	Kavita Iyer

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (52) | university | 00:00:00 | 10 rows

```sql
-- 3. RIGHT JOIN: List all teachers and their assigned students (if any)
SELECT t.teacher_name, s.name AS student_name
FROM students s
RIGHT JOIN teachers t ON s.teacher_id = t.teacher_id;
```

	teacher_name	student_name
1	Rajesh Sharma	Aarav Kapoor
2	Anil Kumar	Ishaan Malhotra
3	Priya Mehta	Riya Sharma
4	Amitabh Joshi	Ananya Gupta
5	Neha Reddy	Kabir Khanna
6	Arun Mishra	Devansh Nair
7	Pooja Patel	Aryan Sinha
8	Ravi Verma	Tara Menon
9	Sanjay Gupta	Neel Patel
10	Kavita Iyer	Sanya Kapoor

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (52) | university | 00:00:00 | 10 rows

```sql
-- 4. FULL OUTER JOIN: List all students and teachers, even if not matched
SELECT s.name AS student_name, t.teacher_name
FROM students s
FULL OUTER JOIN teachers t ON s.teacher_id = t.teacher_id;
```

	student_name	teacher_name
1	Aarav Kapoor	Rajesh Sharma
2	Ishaan Malhotra	Anil Kumar
3	Riya Sharma	Priya Mehta
4	Ananya Gupta	Amitabh Joshi
5	Kabir Khanna	Neha Reddy
6	Devansh Nair	Arun Mishra
7	Aryan Sinha	Pooja Patel
8	Tara Menon	Ravi Verma
9	Neel Patel	Sanjay Gupta
10	Sanya Kapoor	Kavita Iyer

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (52) | university | 00:00:00 | 10 rows

```sql
-- 5. CROSS JOIN: All possible student-teacher pairs
SELECT s.name AS student_name, t.teacher_name
FROM students s
CROSS JOIN teachers t;
```

	student_name	teacher_name
1	Aarav Kapoor	Rajesh Sharma
2	Ishaan Malhotra	Rajesh Sharma
3	Riya Sharma	Rajesh Sharma
4	Ananya Gupta	Rajesh Sharma
5	Kabir Khanna	Rajesh Sharma
6	Devansh Nair	Rajesh Sharma
7	Aryan Sinha	Rajesh Sharma
8	Tara Menon	Rajesh Sharma
9	Neel Patel	Rajesh Sharma
10	Sanya Kapoor	Rajesh Sharma

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (52) | university | 00:00:00 | 100 rows

```sql
-- 6. INNER JOIN using course_id instead of teacher_id
SELECT s.name, s.course_id, t.teacher_name
FROM students s
INNER JOIN teachers t ON s.course_id = t.course_id;
```

	name	course_id	teacher_name
1	Aarav Kapoor	CS101	Rajesh Sharma
2	Ishaan Malhotra	ME102	Anil Kumar
3	Riya Sharma	EC103	Priya Mehta
4	Ananya Gupta	BT104	Amitabh Joshi
5	Kabir Khanna	CE105	Neha Reddy
6	Devansh Nair	IT106	Arun Mishra
7	Aryan Sinha	EE107	Pooja Patel
8	Tara Menon	CH108	Ravi Verma
9	Neel Patel	MA109	Sanjay Gupta
10	Sanya Kapoor	PH110	Kavita Iyer

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (52) | university | 00:00:00 | 10 rows

```sql
-- 7. LEFT JOIN with CASE: Show 'Unassigned' if teacher is NULL
SELECT s.name,
```

```sql
CASE
WHEN t.teacher_name IS NULL THEN 'Unassigned'
ELSE t.teacher_name
END AS teacher_assigned
FROM students s
LEFT JOIN teachers t ON s.teacher_id = t.teacher_id;
```

	name	teacher_assigned
1	Aarav Kapoor	Rajesh Sharma
2	Ishaan Malhotra	Anil Kumar
3	Riya Sharma	Priya Mehta
4	Ananya Gupta	Amitabh Joshi
5	Kabir Khanna	Neha Reddy
6	Devansh Nair	Arun Mishra
7	Aryan Sinha	Pooja Patel
8	Tara Menon	Ravi Verma
9	Neel Patel	Sanjay Gupta
10	Sanya Kapoor	Kavita Iyer

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (52) | university | 00:00:00 | 10 rows

```sql
-- 8. INNER JOIN to get teachers who have at least one student
SELECT DISTINCT t.teacher_name
FROM teachers t
JOIN students s ON s.teacher_id = t.teacher_id;
```

	teacher_name
1	Amitabh Joshi
2	Anil Kumar
3	Arun Mishra
4	Kavita Iyer
5	Neha Reddy
6	Pooja Patel
7	Priya Mehta
8	Rajesh Sharma
9	Ravi Verma
10	Sanjay Gupta

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (52) | university | 00:00:00 | 10 rows

```sql
-- 9. LEFT JOIN: Count how many students are assigned to each teacher
SELECT t.teacher_name, COUNT(s.roll_no) AS student_count
FROM teachers t
LEFT JOIN students s ON t.teacher_id = s.teacher_id
GROUP BY t.teacher_name;
```

	teacher_name	student_count
1	Amitabh Joshi	1
2	Anil Kumar	1
3	Arun Mishra	1
4	Kavita Iyer	1
5	Neha Reddy	1
6	Pooja Patel	1
7	Priya Mehta	1
8	Rajesh Sharma	1
9	Ravi Verma	1
10	Sanjay Gupta	1

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (52) | university | 00:00:00 | 10 rows

```sql
-- 10. INNER JOIN with WHERE condition
SELECT s.name, t.teacher_name
FROM students s
INNER JOIN teachers t ON s.teacher_id = t.teacher_id
WHERE s.admission_status = 'yes';
```

	name	teacher_name
1	Aarav Kapoor	Rajesh Sharma
2	Ishaan Malhotra	Anil Kumar
3	Ananya Gupta	Amitabh Joshi
4	Devansh Nair	Arun Mishra
5	Tara Menon	Ravi Verma
6	Sanya Kapoor	Kavita Iyer

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (52) | university | 00:00:00 | 6 rows

```sql
-- 11. LEFT JOIN to find all students' courses and who is teaching
SELECT s.name, s.course_id, t.teacher_name
FROM students s
LEFT JOIN teachers t ON s.course_id = t.course_id;
```

	name	course_id	teacher_name
1	Aarav Kapoor	CS101	Rajesh Sharma
2	Ishaan Malhotra	ME102	Anil Kumar
3	Riya Sharma	EC103	Priya Mehta
4	Ananya Gupta	BT104	Amitabh Joshi
5	Kabir Khanna	CE105	Neha Reddy
6	Devansh Nair	IT106	Arun Mishra
7	Aryan Sinha	EE107	Pooja Patel
8	Tara Menon	CH108	Ravi Verma
9	Neel Patel	MA109	Sanjay Gupta
10	Sanya Kapoor	PH110	Kavita Iyer

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (52) | university | 00:00:00 | 10 rows

```sql
-- 12. RIGHT JOIN with additional columns
SELECT t.teacher_name, t.course_id, s.name AS student_name
FROM students s
RIGHT JOIN teachers t ON s.teacher_id = t.teacher_id;
```

	teacher_name	course_id	student_name
1	Rajesh Sharma	CS101	Aarav Kapoor
2	Anil Kumar	ME102	Ishaan Malhotra
3	Priya Mehta	EC103	Riya Sharma
4	Amitabh Joshi	BT104	Ananya Gupta
5	Neha Reddy	CE105	Kabir Khanna
6	Arun Mishra	IT106	Devansh Nair
7	Pooja Patel	EE107	Aryan Sinha
8	Ravi Verma	CH108	Tara Menon
9	Sanjay Gupta	MA109	Neel Patel
10	Kavita Iyer	PH110	Sanya Kapoor

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (52) | university | 00:00:00 | 10 rows

```sql
-- 13. FULL JOIN with max_cgpa
SELECT s.name, t.teacher_name, s.max_cgpa
FROM students s
FULL OUTER JOIN teachers t ON s.teacher_id = t.teacher_id;
```

	name	teacher_name	max_cgpa
1	Aarav Kapoor	Rajesh Sharma	10
2	Ishaan Malhotra	Anil Kumar	10
3	Riya Sharma	Priya Mehta	10
4	Ananya Gupta	Amitabh Joshi	10
5	Kabir Khanna	Neha Reddy	10
6	Devansh Nair	Arun Mishra	10
7	Aryan Sinha	Pooja Patel	10
8	Tara Menon	Ravi Verma	10
9	Neel Patel	Sanjay Gupta	10
10	Sanya Kapoor	Kavita Iyer	10

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (52) | university | 00:00:00 | 10 rows

```sql
-- 14. JOIN with alias and ORDER BY
SELECT s.name AS Student, t.teacher_name AS Teacher
FROM students s
JOIN teachers t ON s.teacher_id = t.teacher_id
ORDER BY s.name;
```

	Student	Teacher
1	Aarav Kapoor	Rajesh Sharma
2	Ananya Gupta	Amitabh Joshi
3	Aryan Sinha	Pooja Patel
4	Devansh Nair	Arun Mishra
5	Ishaan Malhotra	Anil Kumar
6	Kabir Khanna	Neha Reddy
7	Neel Patel	Sanjay Gupta
8	Riya Sharma	Priya Mehta
9	Sanya Kapoor	Kavita Iyer
10	Tara Menon	Ravi Verma

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (52) | university | 00:00:00 | 10 rows

```sql
-- 15. CROSS JOIN with condition (avoid full cartesian output)
SELECT s.name AS Student, t.teacher_name AS Teacher
FROM students s
CROSS JOIN teachers t
WHERE s.roll_no % 2 = t.teacher_id % 2;
```

	Student	Teacher
1	Aarav Kapoor	Rajesh Sharma
2	Riya Sharma	Rajesh Sharma
3	Kabir Khanna	Rajesh Sharma
4	Aryan Sinha	Rajesh Sharma
5	Neel Patel	Rajesh Sharma
6	Ishaan Malhotra	Anil Kumar
7	Ananya Gupta	Anil Kumar
8	Devansh Nair	Anil Kumar
9	Tara Menon	Anil Kumar
10	Sanya Kapoor	Anil Kumar

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (52) | university | 00:00:00 | 50 rows

10. Perform at least 10 queries related to sub-queries and corelated sub-queries.

```
USE university

-- 1. Subquery: List students whose CGPA_10 is above the
average
SELECT name, cgpa_10
FROM students
WHERE cgpa_10 > (SELECT AVG(cgpa_10) FROM students WHERE
cgpa_10 IS NOT NULL);
```

	name	cgpa_10
1	Aarav Kapoor	9.2
2	Devansh Nair	9
3	Sanya Kapoor	8.7

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (52) | university | 00:00:00 | 3 rows

```
-- 2. Subquery: Find names of teachers who teach a student
named 'Aarav Kapoor'
SELECT teacher_name
FROM teachers
WHERE teacher_id IN (
SELECT teacher_id FROM students WHERE name = 'Aarav Kapoor'
);
```

	teacher_name
1	Rajesh Sharma

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (52) | university | 00:00:00 | 1 rows

```
-- 3. Subquery: Get the list of students whose course is being
taught by teachers with ID less than 5
SELECT name, course_id
FROM students
WHERE course_id IN (
SELECT course_id FROM teachers WHERE teacher_id < 5
);
```

	name	course_id
1	Aarav Kapoor	CS101
2	Ishaan Malhotra	ME102
3	Riya Sharma	EC103
4	Ananya Gupta	BT104

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (52) | university | 00:00:00 | 4 rows

```
-- 4. Correlated Subquery: Find students whose CGPA_10 is the
highest among students enrolled in the same course
SELECT name, course_id, cgpa_10
FROM students s1
```

```
WHERE cgpa_10 = (
SELECT MAX(cgpa_10) FROM students s2 WHERE s1.course_id =
s2.course_id
);
```

	name	course_id	cgpa_10
1	Sanya Kapoor	PH110	8.7
2	Ishaan Malhotra	ME102	8.1
3	Devansh Nair	IT106	9
4	Aarav Kapoor	CS101	9.2
5	Tara Menon	CH108	7.8
6	Ananya Gupta	BT104	7.5

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (52) | university | 00:00:00 | 6 rows

```
-- 5. Subquery: List names of students who are taught by
'Rajesh Sharma'
SELECT name
FROM students
WHERE teacher_id = (
SELECT teacher_id FROM teachers WHERE teacher_name = 'Rajesh
Sharma'
);
```

	name
1	Aarav Kapoor

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (52) | university | 00:00:00 | 1 rows

```
-- 6. Correlated Subquery: Find students with the highest
CGPA_10 under each teacher
SELECT name, cgpa_10, teacher_id
FROM students s1
WHERE cgpa_10 = (
SELECT MAX(cgpa_10)
FROM students s2
WHERE s1.teacher_id = s2.teacher_id
);
```

	name	cgpa_10	teacher_id
1	Sanya Kapoor	8.7	10
2	Tara Menon	7.8	8
3	Devansh Nair	9	6
4	Ananya Gupta	7.5	4
5	Ishaan Malh...	8.1	2
6	Aarav Kapoor	9.2	1

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (52) | university | 00:00:00 | 6 rows

```
-- 7. Subquery in SELECT clause: Show each student's CGPA and
the average CGPA_10 of all students
SELECT name, cgpa_10,
(SELECT AVG(cgpa_10) FROM students WHERE cgpa_10 IS NOT NULL)
AS avg_cgpa
FROM students;
```

	name	cgpa_10	avg_cgpa
1	Aarav Kapoor	9.2	8.38333333333333
2	Ishaan Malhotra	8.1	8.38333333333333
3	Riya Sharma	NULL	8.38333333333333
4	Ananya Gupta	7.5	8.38333333333333
5	Kabir Khanna	NULL	8.38333333333333
6	Devansh Nair	9	8.38333333333333
7	Aryan Sinha	NULL	8.38333333333333
8	Tara Menon	7.8	8.38333333333333
9	Neel Patel	NULL	8.38333333333333
10	Sanya Kapoor	8.7	8.38333333333333

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (52) | university | 00:00:00 | 10 rows

```sql
-- 8. Subquery: Find students who are in a course taught by a teacher whose name starts with 'P'
SELECT name
FROM students
WHERE course_id IN (
SELECT course_id FROM teachers WHERE teacher_name LIKE 'P%'
);
```

	name
1	Riya Sharma
2	Aryan Sinha

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (52) | university | 00:00:00 | 2 rows

```sql
-- 9. Correlated Subquery: Find students whose CGPA_12 is higher than the CGPA_12 of any student in a different course.
SELECT name, cgpa_12, course_id
FROM students s1
WHERE cgpa_12 > (
SELECT MAX(cgpa_12)
FROM students s2
WHERE s1.course_id != s2.course_id AND s2.cgpa_12 IS NOT NULL
);
```

	name	cgpa_12	course_id
1	Devansh Nair	9.3	IT106

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (52) | university | 00:00:00 | 1 rows

```sql
-- 10. Subquery: List all teachers who teach at least one student with CGPA_10 above 9
SELECT DISTINCT teacher_name
FROM teachers
WHERE teacher_id IN (
SELECT teacher_id FROM students WHERE cgpa_10 > 9
);
```

	teacher_name
1	Rajesh Sharma

Query executed successfully. | DEN\SQLEXPRESS (16.0 RTM) | DEN\aaumg (52) | university | 00:00:00 | 1 rows

11. Consider the following Schema Student(Sid Pk, Sname), Supplier(Sid:Int, Sname:Str, Address:Str), Parts(Pid:Int, Pname:Str, Color:Str), Catalog(Sid, Pid, Cost:Real). Create tables and insert at least 10 Entries in each table and perform 5 Queries.

```
-- Create a new database
CREATE DATABASE PartsDatabase;
GO

-- Use the database
USE PartsDatabase;
GO

-- Create Student table
CREATE TABLE Student (
    Sid INT PRIMARY KEY,
    Sname VARCHAR(50)
);

-- Create Supplier table
CREATE TABLE Supplier (
    Sid INT PRIMARY KEY,
    Sname VARCHAR(50),
    Address VARCHAR(100)
);

-- Create Parts table
CREATE TABLE Parts (
    Pid INT PRIMARY KEY,
    Pname VARCHAR(50),
    Color VARCHAR(30)
);

-- Create Catalog table
CREATE TABLE Catalog (
    Sid INT,
    Pid INT,
    Cost REAL,
    FOREIGN KEY (Sid) REFERENCES Supplier(Sid),
    FOREIGN KEY (Pid) REFERENCES Parts(Pid)
);

-- Insert values into Student
INSERT INTO Student VALUES
(1, 'Aarav'),
```

```
(2, 'Riya'),
(3, 'Ishaan'),
(4, 'Ananya'),
(5, 'Kabir'),
(6, 'Tara'),
(7, 'Dev'),
(8, 'Sanya'),
(9, 'Neel'),
(10, 'Kriti');

-- Insert values into Supplier
INSERT INTO Supplier VALUES
(201, 'AutoParts Ltd', 'Delhi'),
(202, 'GearTech', 'Mumbai'),
(203, 'RoboSupplies', 'Pune'),
(204, 'SteelWorks', 'Kolkata'),
(205, 'ToolChain', 'Chennai'),
(206, 'NanoParts', 'Bangalore'),
(207, 'MechMart', 'Ahmedabad'),
(208, 'QuickFix', 'Hyderabad'),
(209, 'BuildCore', 'Lucknow'),
(210, 'PartsDepot', 'Jaipur');

-- Insert values into Parts
INSERT INTO Parts VALUES
(301, 'Screw', 'Silver'),
(302, 'Bolt', 'Black'),
(303, 'Nut', 'Black'),
(304, 'Washer', 'Silver'),
(305, 'Gear', 'Gray'),
(306, 'Sensor', 'White'),
(307, 'Switch', 'Red'),
(308, 'Fan', 'White'),
(309, 'Motor', 'Black'),
(310, 'Rotor', 'Gray');

-- Insert values into Catalog
INSERT INTO Catalog VALUES
(201, 301, 5.5),
(202, 302, 4.0),
(203, 303, 3.0),
(204, 304, 6.0),
(205, 305, 10.0),
(206, 306, 25.0),
(207, 307, 15.0),
(208, 308, 20.0),
(209, 309, 18.0),
(210, 310, 12.5);
```

```sql
-- ----------------------------
-- Perform 5 Sample Queries
-- ----------------------------

-- 1. List all parts that are black in color
SELECT * FROM Parts
WHERE Color = 'Black';

-- 2. List supplier names along with the parts they supply
SELECT S.Sname AS SupplierName, P.Pname AS PartName, C.Cost
FROM Supplier S
JOIN Catalog C ON S.Sid = C.Sid
JOIN Parts P ON P.Pid = C.Pid;

-- 3. Display all student names that start with 'A'
SELECT * FROM Student
WHERE Sname LIKE 'A%';

-- 4. Show parts with cost greater than 10
SELECT P.Pname, C.Cost
FROM Catalog C
JOIN Parts P ON P.Pid = C.Pid
WHERE C.Cost > 10;

-- 5. Count how many parts are supplied by each supplier
SELECT S.Sname, COUNT(C.Pid) AS PartCount
FROM Supplier S
JOIN Catalog C ON S.Sid = C.Sid
GROUP BY S.Sname
ORDER BY PartCount DESC;
```

	Pid	Pname	Color
1	302	Bolt	Black
2	303	Nut	Black
3	309	Motor	Black

	SupplierName	PartName	Cost
1	AutoParts Ltd	Screw	5.5
2	GearTech	Bolt	4
3	RoboSupplies	Nut	3
4	SteelWorks	Washer	6
5	ToolChain	Gear	10
6	NanoParts	Sensor	25
7	MechMart	Switch	15
8	QuickFix	Fan	20
9	BuildCore	Motor	18
10	PartsDepot	Rotor	12.5

	Sid	Sname
1	1	Aarav
2	4	Ananya

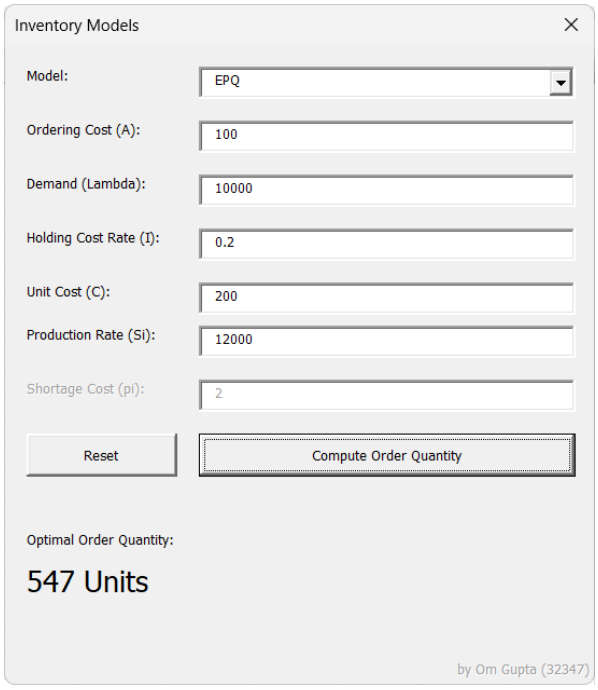
	Pname	Cost
1	Sensor	25
2	Switch	15
3	Fan	20
4	Motor	18
5	Rotor	12.5

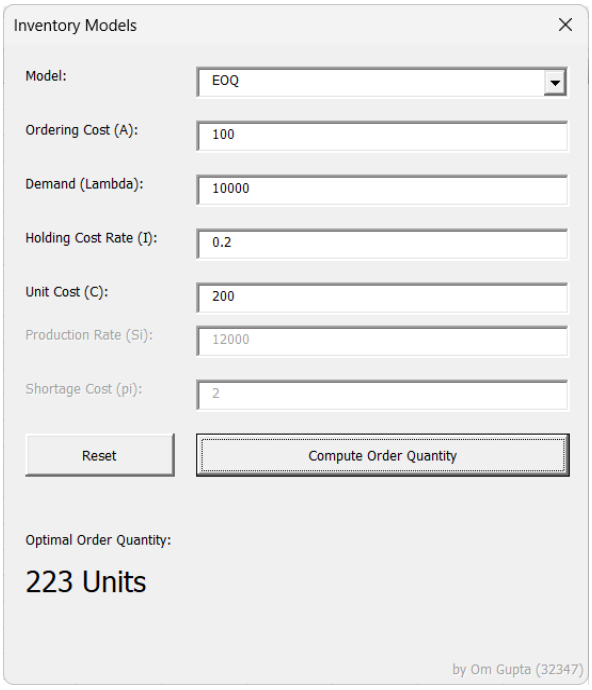
	Sname	PartCount
1	AutoParts Ltd	1
2	BuildCore	1
3	GearTech	1
4	MechMart	1
5	NanoParts	1
6	PartsDepot	1
7	QuickFix	1
8	RoboSuppli...	1
9	SteelWorks	1
10	ToolChain	1

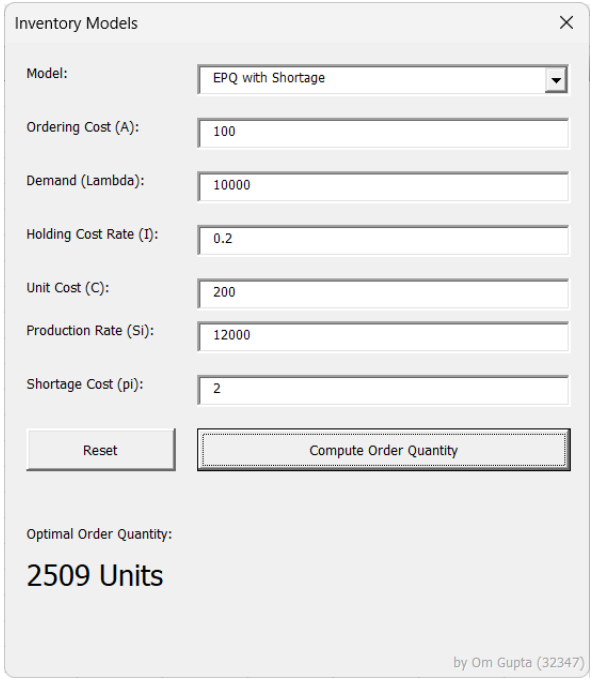
Query executed successfully. DEN\SQLEXPRESS (16.0 RTM) DEN\aaumg (64) PartsDatabase 00:00:00 30 rows

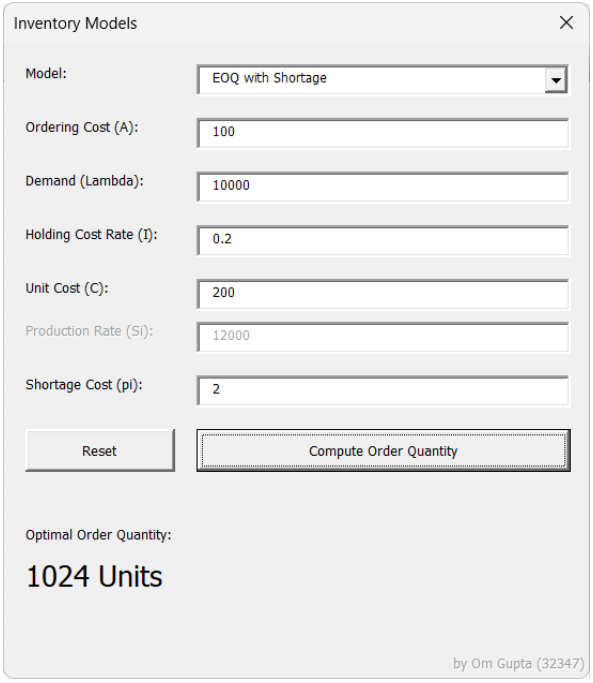
12. Create a form in visual basic to implement economic order quantity models (4).

Screenshots:









VBA Script:

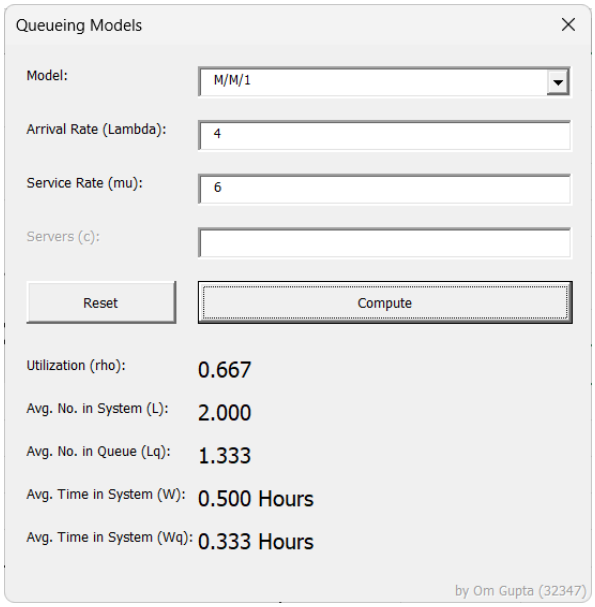
```
Private Sub Compute_Click()
    Dim model As String
    Dim A As Double
    Dim lam As Double
```

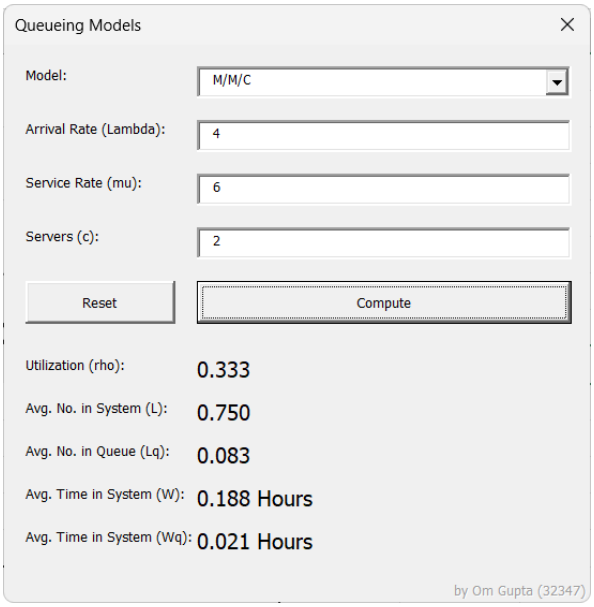
```
    Dim I As Double
    Dim C As Double
    Dim Si As Double
    Dim Pi As Double
    Dim qty As Double

    If Trim(A_txt.Value) = "" Then A = 0 Else A = CDbl(A_txt.Value)
    If Trim(lam_txt.Value) = "" Then lam = 0 Else lam =
CDbl(lam_txt.Value)
    If Trim(I_txt.Value) = "" Then I = 0 Else I = CDbl(I_txt.Value)
    If Trim(C_txt.Value) = "" Then C = 0 Else C = CDbl(C_txt.Value)

    If Trim(Si_txt.Value) = "" Then Si = 0 Else Si =
CDbl(Si_txt.Value)
    If Trim(Pi_txt.Value) = "" Then Pi = 0 Else Pi =
CDbl(Pi_txt.Value)

    model = ModelComboBox.Value

    If model = "EOQ" Then
        qty = Int(Sqr((2 * A * lam) / (I * C)))
    ElseIf model = "EOQ with Shortage" Then
        qty = Int(Sqr((2 * A * lam * (Pi + I * C)) / (I * C * Pi)))
    ElseIf model = "EPQ" Then
        qty = Int(Sqr((2 * A * lam * Si) / (I * C * (Si - lam))))
    ElseIf model = "EPQ with Shortage" Then
        qty = Int(Sqr((2 * A * lam * (Pi + I * C) * Si) / (I * C *
Pi * (Si - lam))))
    Else: qty = 0
    End If

    qty_txt.Caption = Int(qty) & " Units"
End Sub

Private Sub ModelComboBox_Change()
    Compute.Enabled = True
    Reset.Enabled = True
    qty_txt.Caption = "0 Units"

    Dim model As String
    model = ModelComboBox.Value
    If model = "EOQ" Then
        Si_txt.Enabled = False
        Si_label.Enabled = False
        Pi_txt.Enabled = False
        Pi_label.Enabled = False
    ElseIf model = "EOQ with Shortage" Then
        Si_txt.Enabled = False
        Si_label.Enabled = False
```

```
        Pi_txt.Enabled = True
        Pi_label.Enabled = True
    ElseIf model = "EPQ" Then
        Si_txt.Enabled = True
        Si_label.Enabled = True
        Pi_txt.Enabled = False
        Pi_label.Enabled = False
    ElseIf model = "EPQ with Shortage" Then
        Si_txt.Enabled = True
        Si_label.Enabled = True
        Pi_txt.Enabled = True
        Pi_label.Enabled = True
    Else:
        Si_txt.Enabled = False
        Si_label.Enabled = False
        Pi_txt.Enabled = False
        Pi_label.Enabled = False
    End If

End Sub

Private Sub Reset_Click()
    A_txt.Value = ""
    lam_txt.Value = ""
    I_txt.Value = ""
    C_txt.Value = ""
    Si_txt.Value = ""
    Pi_txt.Value = ""

    ModelComboBox.Text = "Select"
    qty_txt.Caption = "0 Units"

    Si_txt.Enabled = False
    Si_label.Enabled = False
    Pi_txt.Enabled = False
    Pi_label.Enabled = False
    Compute.Enabled = False
    Reset.Enabled = False
End Sub

Private Sub UserForm_Initialize()
ModelComboBox.Clear
With ModelComboBox
    .AddItem "EOQ"
    .AddItem "EOQ with Shortage"
    .AddItem "EPQ"
    .AddItem "EPQ with Shortage"
End With
End Sub
```

13. Create a form in visual basic to implement M|M|1 and M|M|C models to calculate all its characteristics.

Screenshots:





VBA Script:

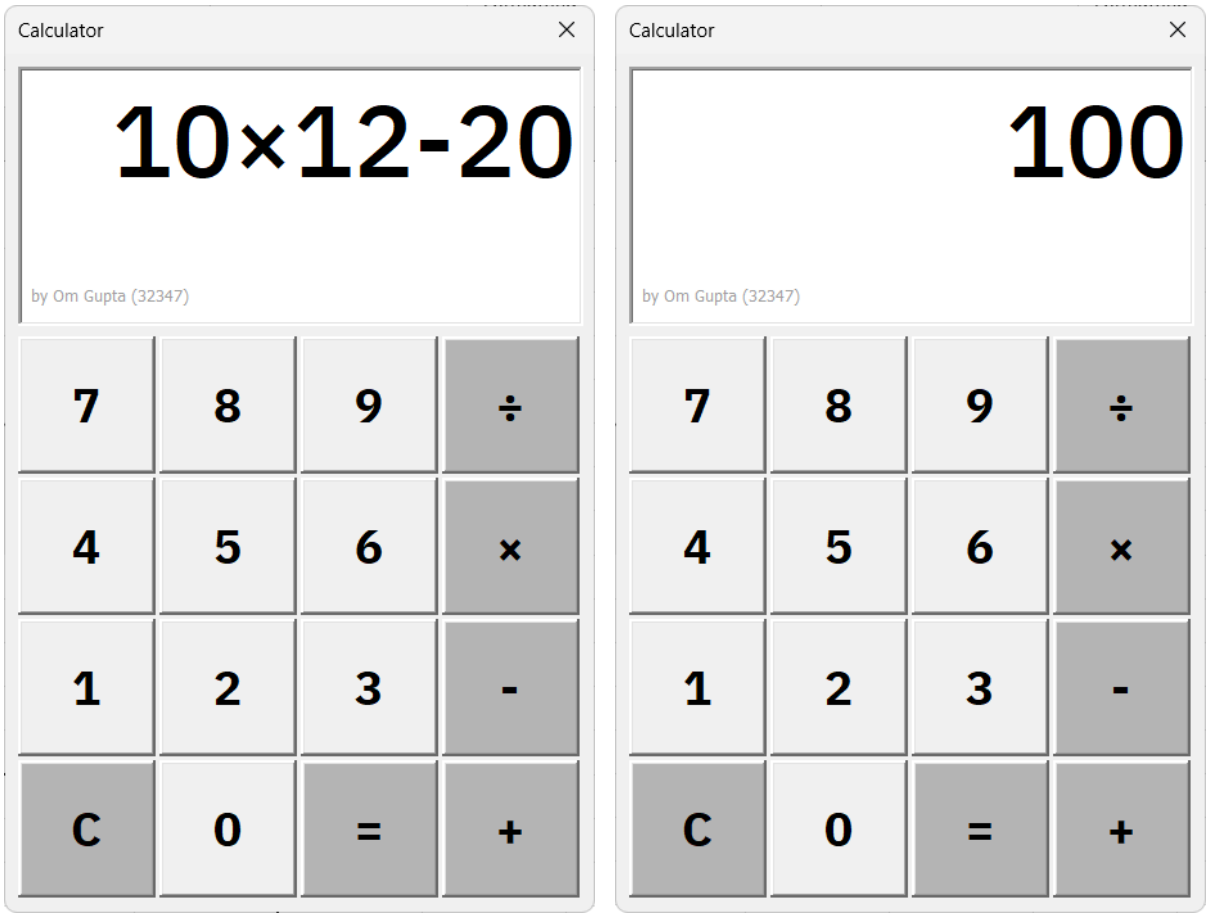
```
Private Sub UserForm_Initialize()
    ModelComboBox.Clear
    With ModelComboBox
        .AddItem "M/M/1"
        .AddItem "M/M/C"
    End With
    c_label.Enabled = False
    c_txt.Enabled = False
    Compute.Enabled = False
    Reset.Enabled = False
End Sub

Private Sub ModelComboBox_Change()
    Compute.Enabled = True
    Reset.Enabled = True
    If ModelComboBox.Value = "M/M/C" Then
        c_label.Enabled = True
        c_txt.Enabled = True
    Else
        c_label.Enabled = False
        c_txt.Enabled = False
    End If
End Sub

Private Sub Compute_Click()
```

```
    Dim lam As Double, mu As Double, c As Integer
    Dim rho As Double, L As Double, Lq As Double, W As Double, Wq As
Double
    Dim model As String

    lam = Val(lam_txt.Value)
    mu = Val(mu_txt.Value)
    model = ModelComboBox.Value

    If model = "M/M/1" Then
        If mu <= lam Then
            MsgBox "Invalid values: System is unstable"
            Exit Sub
        End If
        rho = lam / mu
        L = rho / (1 - rho)
        Lq = rho ^ 2 / (1 - rho)
        W = 1 / (mu - lam)
        Wq = lam / (mu * (mu - lam))
    ElseIf model = "M/M/C" Then
        c = Val(c_txt.Value)
        If mu * c <= lam Then
            MsgBox "Invalid values: System is unstable"
            Exit Sub
        End If
        rho = lam / (c * mu)

        ' Calculate P0
        Dim sum As Double: sum = 0
        Dim i As Integer
        For i = 0 To c - 1
            sum = sum + (lam / mu) ^ i / WorksheetFunction.Fact(i)
        Next i
        Dim lastTerm As Double
        lastTerm = ((lam / mu) ^ c) / (WorksheetFunction.Fact(c) *
(1 - rho))
        Dim P0 As Double: P0 = 1 / (sum + lastTerm)

        Lq = (P0 * ((lam / mu) ^ c) * rho) /
(WorksheetFunction.Fact(c) * ((1 - rho) ^ 2))
        L = Lq + lam / mu
        Wq = Lq / lam
        W = L / lam
    End If

    ' Output
    rho_label.Caption = Format(rho, "0.000")
    L_label.Caption = Format(L, "0.000")
    Lq_label.Caption = Format(Lq, "0.000")
```

```
    W_label.Caption = Format(W, "0.000") + " Hours"
    Wq_label.Caption = Format(Wq, "0.000") + " Hours"
End Sub

Private Sub Reset_Click()
    lam_txt.Value = ""
    mu_txt.Value = ""
    c_txt.Value = ""

    rho_label.Caption = "0"
    L_label.Caption = "0"
    Lq_label.Caption = "0"
    W_label.Caption = "0"
    Wq_label.Caption = "0"

    ModelComboBox.Text = "Select"
    c_label.Enabled = False
    c_txt.Enabled = False
    Compute.Enabled = False
    Reset.Enabled = False
End Sub
```

14. Create a form in visual basic to simulate a simple calculator.

Screenshots:



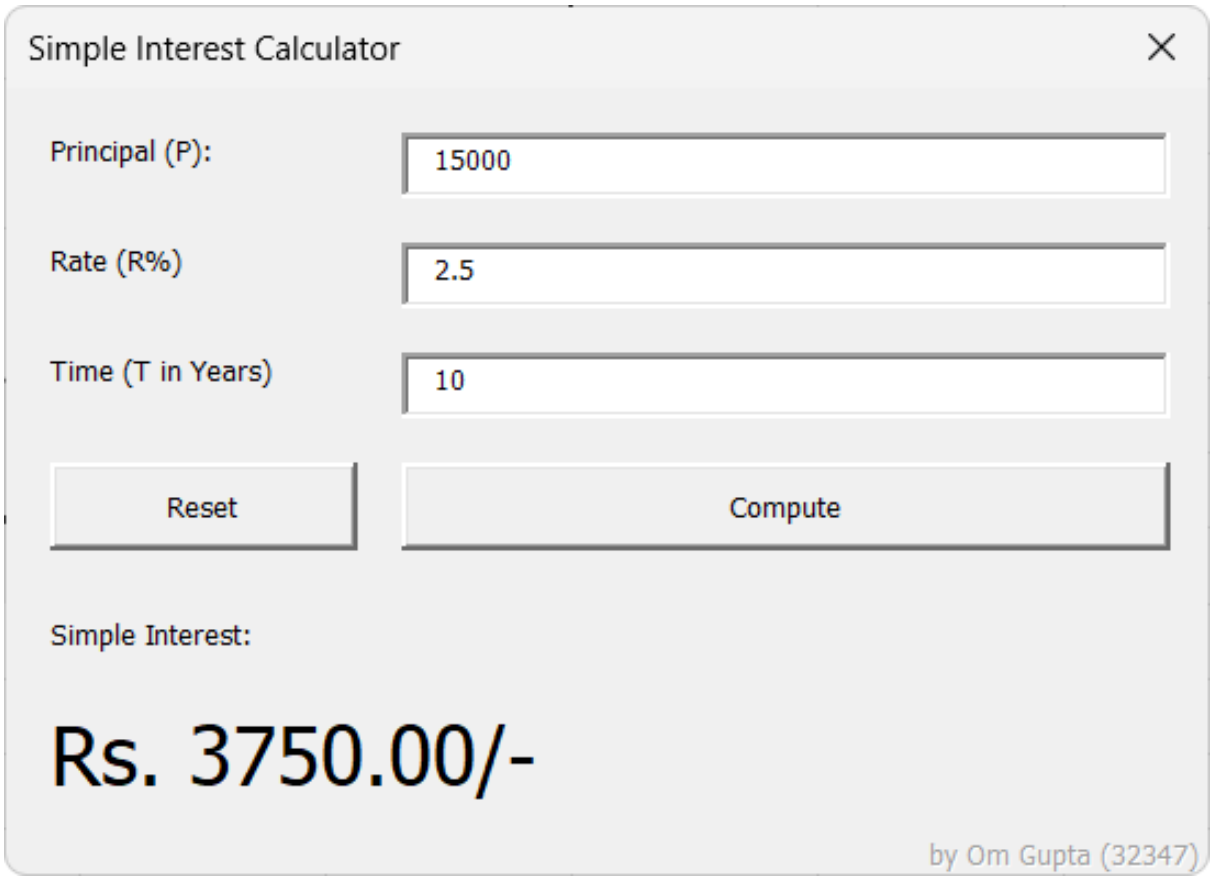
VBA Script:

```
Dim expression As String

Private Sub UserForm_Initialize()
    txtDisplay.Value = ""
    expression = ""
End Sub

Private Sub AppendToExpression(val As String)
    expression = expression & val
    txtDisplay.Value = expression
End Sub

Private Sub btn0_Click(): AppendToExpression "0": End Sub
Private Sub btn1_Click(): AppendToExpression "1": End Sub
Private Sub btn2_Click(): AppendToExpression "2": End Sub
Private Sub btn3_Click(): AppendToExpression "3": End Sub
```

```
Private Sub btn4_Click(): AppendToExpression "4": End Sub
Private Sub btn5_Click(): AppendToExpression "5": End Sub
Private Sub btn6_Click(): AppendToExpression "6": End Sub
Private Sub btn7_Click(): AppendToExpression "7": End Sub
Private Sub btn8_Click(): AppendToExpression "8": End Sub
Private Sub btn9_Click(): AppendToExpression "9": End Sub

Private Sub btnAdd_Click(): AppendToExpression "+": End Sub
Private Sub btnSub_Click(): AppendToExpression "-": End Sub
Private Sub btnMul_Click(): AppendToExpression "×": End Sub
Private Sub btnDiv_Click(): AppendToExpression "÷": End Sub

Private Sub btnClear_Click()
    expression = ""
    txtDisplay.Value = ""
End Sub

Private Sub btnEqual_Click()
    On Error GoTo errHandler
    Dim result As Variant
    result = Evaluate("=" & expression)
    txtDisplay.Value = result
    expression = result
    Exit Sub

errHandler:
    txtDisplay.Value = "Error"
    expression = ""
End Sub
```

15. Create a form in visual basic to calculate simple interest.

Screenshot:



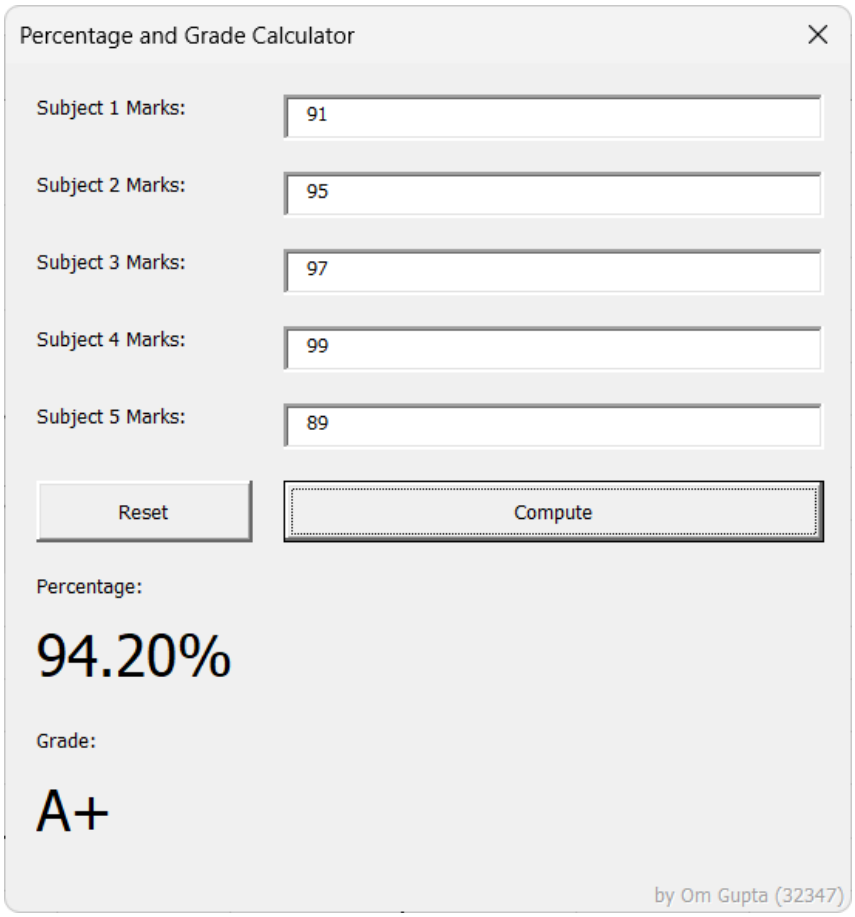
VBA Script:

```
Private Sub btnCalc_Click()
    Dim P As Double, R As Double, T As Double, SI As Double

    ' Validate inputs
    If IsNumeric(txtP.Value) And IsNumeric(txtR.Value) And
IsNumeric(txtT.Value) Then
        P = CDbl(txtP.Value)
        R = CDbl(txtR.Value)
        T = CDbl(txtT.Value)

        ' Calculate Simple Interest
        SI = (P * R * T) / 100
        lblOutput.Caption = "Rs. " & Format(SI, "0.00") & "/-"

    Else
        MsgBox "Please enter valid numeric values for P, R, and T.",
vbExclamation, "Input Error"
```

```
    End If
End Sub

Private Sub btnReset_Click()

    txtP.Value = ""

    txtR.Value = ""

    txtT.Value = ""

    lblOutput.Caption = ""

End Sub

Private Sub UserForm_Initialize()
    lblOutput.Caption = ""

End Sub
```

17. Create a form in Visual Basic to compute percentage and grade of a student.

Screenshot:



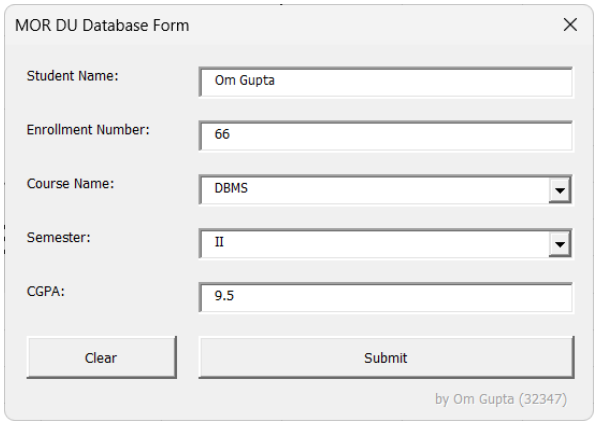
VBA Script:

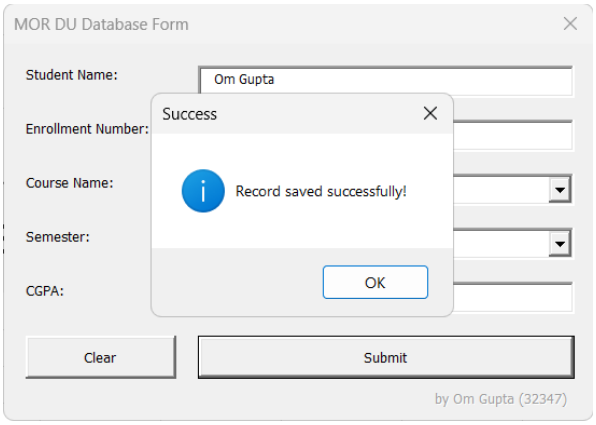
```
Private Sub btnCalculate_Click()
    Dim marks(1 To 5) As Double
    Dim total As Double
    Dim percentage As Double
    Dim i As Integer

    marks(1) = CDbl(txtSub1.Value)
    marks(2) = CDbl(txtSub2.Value)
    marks(3) = CDbl(txtSub3.Value)
    marks(4) = CDbl(txtSub4.Value)
    marks(5) = CDbl(txtSub5.Value)

    For i = 1 To 5
        If marks(i) < 0 Or marks(i) > 100 Then
            MsgBox "Marks should be between 0 and 100.", vbCritical
            Exit Sub
```

```
        End If
        total = total + marks(i)
    Next i

    percentage = total / 5
    lblPercentage.Caption = Format(percentage, "0.00") & "%"

    Dim grade As String
    Select Case percentage
        Case Is >= 90
            grade = "A+"
        Case Is >= 80
            grade = "A"
        Case Is >= 70
            grade = "B"
        Case Is >= 60
            grade = "C"
        Case Is >= 50
            grade = "D"
        Case Else
            grade = "F"
    End Select

    lblGrade.Caption = grade
    Exit Sub

ErrHandler:
    MsgBox "Please enter valid numeric marks.", vbCritical
End Sub

Private Sub btnReset_Click()
    txtName = ""
    txtSub1 = ""
    txtSub2 = ""
    txtSub3 = ""
    txtSub4 = ""
    txtSub5 = ""
End Sub

Private Sub UserForm_Click()

End Sub
```

16. Create a form in visual basic to prepare a database.

Screenshots:





STUDENT NAME	ENROLLMENT NO.	COURSE NAME	SEMESTER	CGPA
Om Gupta	66	DBMS	II	9.5
Meera	30	DBMS	III	8.86
Saanvi	19	Queueing Theory	III	6.4
Vivaan	84	DBMS	II	5.44
Aanya	53	Queueing Theory	IV	8.29
Aarav	73	Convex Optimization	I	8.29
Aryan	77	DBMS	I	7.72
Tanvi	65	Statistics	II	7.58
Diya	41	Queueing Theory	III	6.85
Simran	28	DBMS	III	6.56
Aditya	42	Python Programming	IV	7.77
Tanvi	48	Queueing Theory	IV	7.04
Vivaan	20	Statistics	I	9.42
Amit	69	Python Programming	IV	5.69
Amit	50	Queueing Theory	IV	8.17
Kavya	89	DBMS	III	7.96
Aanya	23	Queueing Theory	III	8.42
Saanvi	11	Convex Optimization	II	7.15
Simran	43	Python Programming	I	6.02
Priya	24	Queueing Theory	II	7.53
Meera	71	DBMS	I	9.67

VBA Script:

```
Private Sub UserForm_Initialize()
    With cmbCourse
        .AddItem "Linear Programming"
```

```
        .AddItem "DBMS"
        .AddItem "Python Programming"
        .AddItem "Statistics"
        .AddItem "Queueing Theory"
        .AddItem "Convex Optimization"
    End With

    cmbSemester.AddItem "I"
    cmbSemester.AddItem "II"
    cmbSemester.AddItem "III"
    cmbSemester.AddItem "IV"
End Sub

Private Sub btnSubmit_Click()
    Dim ws As Worksheet
    Set ws = ThisWorkbook.Sheets("MOR_DB")

    Dim nextRow As Long
    nextRow = ws.Cells(ws.Rows.Count, "A").End(xlUp).Row + 1

    Dim enrollNum As Long
    Dim cgpaVal As Double

    ' Input validation
    If txtName = "" Or txtEnroll = "" Or cmbCourse = "" Or
cmbSemester = "" Or txtCGPA = "" Then
        MsgBox "Please fill in all fields", vbExclamation, "Missing
Data"
        Exit Sub
    End If

    ' Check if Enrollment No. is Integer
    If Not IsNumeric(txtEnroll) Or InStr(txtEnroll, ".") > 0 Then
        MsgBox "Enrollment Number must be an integer.", vbCritical,
"Invalid Input"
        Exit Sub
    Else
        enrollNum = CLng(txtEnroll)
    End If

    ' Check if CGPA is Float (between 0.0 and 10.0)

    If Not IsNumeric(txtCGPA) Then
        MsgBox "CGPA must be a number.", vbCritical, "Invalid Input"
        Exit Sub
    Else
        cgpaVal = CDbl(txtCGPA)
        If cgpaVal < 0 Or cgpaVal > 10 Then
```

```
            MsgBox "CGPA must be between 0.0 and 10.0.", vbCritical,
"Invalid Range"
            Exit Sub
        End If
    End If

    ' Save to sheet

    ws.Cells(nextRow, 1).Value = txtName
    ws.Cells(nextRow, 2).Value = enrollNum
    ws.Cells(nextRow, 3).Value = cmbCourse
    ws.Cells(nextRow, 4).Value = cmbSemester
    ws.Cells(nextRow, 5).Value = cgpaVal

    MsgBox "Record saved successfully!", vbInformation, "Success"
    Call btnClear_Click
End Sub

Private Sub btnClear_Click()
    txtName = ""
    txtEnroll = ""
    cmbCourse = ""
    cmbSemester = ""
    txtCGPA = ""
End Sub
```

Links

1. https://github.com/AumGupta/MOR/blob/main/DBMS/Practicals/Practical%201%20(BookMyShow).sql
2. https://github.com/AumGupta/MOR/blob/main/DBMS/Practicals/Practical%202%20(University).sql
3. https://github.com/AumGupta/MOR/blob/main/DBMS/Practicals/Practical%203%20(Update%20Uni).sql
4. https://github.com/AumGupta/MOR/blob/main/DBMS/Practicals/Practical%204.sql
5. https://github.com/AumGupta/MOR/blob/main/DBMS/Practicals/Practical%205.sql
6. https://github.com/AumGupta/MOR/blob/main/DBMS/Practicals/Practical%206.sql
7. https://github.com/AumGupta/MOR/blob/main/DBMS/Practicals/Practical%207.sql
8. https://github.com/AumGupta/MOR/blob/main/DBMS/Practicals/Practical%208.sql
9. https://github.com/AumGupta/MOR/blob/main/DBMS/Practicals/Practical%209.sql
10. https://github.com/AumGupta/MOR/blob/main/DBMS/Practicals/Practical%2010.sql
11. https://github.com/AumGupta/MOR/blob/main/DBMS/Practicals/Practical%2011.sql
12. https://github.com/AumGupta/MOR/blob/main/DBMS/VBA/Inventory-Models.xlsm
13. https://github.com/AumGupta/MOR/blob/main/DBMS/VBA/Queueing-Models.xlsm
14. https://github.com/AumGupta/MOR/blob/main/DBMS/VBA/Calculator.xlsm
15. https://github.com/AumGupta/MOR/blob/main/DBMS/VBA/Simplet-Interest-Calculator.xlsm
16. https://github.com/AumGupta/MOR/blob/main/DBMS/VBA/Percentage-and-Grade-Calculator.xlsm
17. https://github.com/AumGupta/MOR/blob/main/DBMS/VBA/Database.xlsm