DATABASE PROJECT

Group Members:

Ch. M Arslan Aslam F2022266847

M. Ali Hassan F2022266882

Section: V13

Topic: School Management System

1) Normalization:

DE normalized table of Student Parent:

	Student_ID	Student_Name	DOB	Gender	Age	Admission_Date	Course_Id	Parent_ID	Guardian_ID	Class_ID	Challan_No	Section	Parent_Name	CNIC	Parent_Phone_No	Parent_Address	Parent_Gender
1	S31	Ahmed	2014-04-15	M	10	2023-01-01	C21	P23	G23	13	F23	C	Mr. Ahmed	3456789012345	03211223344	789 Oak St, City	M
2	S32	Hira	2015-02-20	F	9	2023-01-01	C22	P24	G24	14	F24	D	Ms. Fatima	4567890123456	03311223344	901 Pine St, City	F
3	S33	Adnan	2013-07-10	M	11	2023-01-01	C23	P25	G25	15	F25	E	Mr. Zaid	5678901234567	03411223344	234 Maple St, City	M
4	S34	Sana	2012-06-05	F	12	2023-01-01	C24	P26	G26	16	F26	F	Ms. Sara	6789012345678	03511223344	567 Cedar St, City	F
5	S35	Bilal	2011-05-01	M	13	2023-01-01	C25	P27	G27	17	F27	G	Mr. Ali	7890123456789	03611223344	890 Birch St, City	M
6	S36	Madiha	2010-04-03	F	14	2023-01-01	C26	P28	G28	18	F28	H	Ms. Laila	8901234567890	03711223344	123 Walnut St, City	F
7	S37	Omar	2009-03-10	M	15	2023-01-01	C27	P29	G29	19	F29	I	Mr. Imran	9012345678901	03811223344	456 Redwood St, City	M
8	S38	Zoya	2008-02-15	F	16	2023-01-01	C28	P30	G30	20	F30	J	Ms. Zara	0123456789012	03911223344	789 Sycamore St, City	F
9	S39	Raza	2007-01-20	M	17	2023-01-01	C29	P21	G21	11	F21	A	Mr. Kareem	1234567890123	03011223344	123 Main St, City	M
10	S40	Aisha	2006-12-25	F	18	2023-01-01	C30	P22	G22	12	F22	B	Ms. Ayesha	2345678901234	03111223344	456 Elm St, City	F

Normalize up to 3rd Normal form.

Student:

Results Messages

	Student_ID	Student_Name	DOB	Gender	Age	Admission_Date	Course_Id	Parent_ID	Guardian_ID	Class_ID	Challan_No	Section
1	S21	Ali	2015-03-01	M	9	2023-01-01	C21	P21	G21	11	F21	A
2	S22	Sara	2014-05-10	F	10	2023-01-01	C22	P22	G22	12	F22	B
3	S23	Arslan	2013-08-15	M	11	2023-01-01	C23	P23	G23	13	F23	C
4	S24	Fatima	2012-11-20	F	12	2023-01-01	C24	P24	G24	14	F24	D
5	S25	Zaid	2011-02-25	M	13	2023-01-01	C25	P25	G25	15	F25	E
6	S26	Ayesha	2010-04-30	F	14	2023-01-01	C26	P26	G26	16	F26	F
7	S27	Salman	2009-07-05	M	15	2023-01-01	C27	P27	G27	17	F27	G
8	S28	Laila	2008-09-10	F	16	2023-01-01	C28	P28	G28	18	F28	H
9	S29	Imran	2007-12-15	M	17	2023-01-01	C29	P29	G29	19	F29	I
10	S30	Zara	2006-02-20	F	18	2023-01-01	C30	P30	G30	20	F30	J

Parent:

Results Messages

	Parent_ID	Parent_Name	CNIC	Phone_No	Address	Gender	Email	P_JOB	Spouse
1	P21	Mr. Kareem	1234567890123	03011223344	123 Main St, City	M	kareem@example.com	Engineer	Ayesha
2	P22	Ms. Ayesha	2345678901234	03111223344	456 Elm St, City	F	ayesha@example.com	Doctor	Kareem
3	P23	Mr. Ahmed	3456789012345	03211223344	789 Oak St, City	M	ahmed@example.com	Lawyer	Fatima
4	P24	Ms. Fatima	4567890123456	03311223344	901 Pine St, City	F	fatima@example.com	Artist	Ahmed
5	P25	Mr. Zaid	5678901234567	03411223344	234 Maple St, City	M	zaid@example.com	Professor	Sara
6	P26	Ms. Sara	6789012345678	03511223344	567 Cedar St, City	F	sara@example.com	Pilot	Zaid
7	P27	Mr. Ali	7890123456789	03611223344	890 Birch St, City	M	ali@example.com	Journalist	Laila
8	P28	Ms. Laila	8901234567890	03711223344	123 Walnut St, City	F	laila@example.com	Architect	Ali
9	P29	Mr. Imran	9012345678901	03811223344	456 Redwood St, City	M	imran@example.com	Engineer	Zara
10	P30	Ms. Zara	0123456789012	03911223344	789 Sycamore St, City	F	zara@example.com	Doctor	Imran

Class:

Results | Messages

	Class_ID	Section	Class_Title	C_Total_Students	Age_Group	Gender_Group
1	11	A	Grade 1	20	6-7	Mixed
2	12	B	Grade 2	22	7-8	Mixed
3	13	C	Grade 3	24	8-9	Mixed
4	14	D	Grade 4	25	9-10	Mixed
5	15	E	Grade 5	26	10-11	Mixed
6	16	F	Grade 6	28	11-12	Mixed
7	17	G	Grade 7	30	12-13	Mixed
8	18	H	Grade 8	32	13-14	Mixed
9	19	I	Grade 9	34	14-15	Mixed
10	20	J	Grade 10	36	15-16	Mixed

Course:

Results | Messages

	Course_Id	Course_title	Grade	Instructor
1	C21	English Literature	A	Mr. Salman
2	C22	Computer Science	A	Ms. Warda
3	C23	Economics	A	Mr. Moazzam
4	C24	Business Management	A	Ms. Mahibah
5	C25	Chemistry	A	Mr. Waqas
6	C26	Fine Arts	A	Ms. Maham
7	C27	Music	A	Mr. Daud
8	C28	Psychology	A	Ms. Saira
9	C29	Sociology	A	Mr. Hussain
10	C30	Health Education	A	Ms. Aleena

Guardian:

Results | Messages

	Guardian_ID	Guardian_Name	CNIC	Phone_No	Address	Gender	Email	Relation
1	G21	Mr. Kareem	1234567890123	03011223344	123 Main St, City	M	kareem@example.com	Father
2	G22	Ms. Ayesha	2345678901234	03111223344	456 Elm St, City	F	ayesha@example.com	Mother
3	G23	Mr. Ahmed	3456789012345	03211223344	789 Oak St, City	M	ahmed@example.com	Father
4	G24	Ms. Fatima	4567890123456	03311223344	901 Pine St, City	F	fatima@example.com	Mother
5	G25	Mr. Zaid	5678901234567	03411223344	234 Maple St, City	M	zaid@example.com	Father
6	G26	Ms. Sara	6789012345678	03511223344	567 Cedar St, City	F	sara@example.com	Mother
7	G27	Mr. Ali	7890123456789	03611223344	890 Birch St, City	M	ali@example.com	Father
8	G28	Ms. Laila	8901234567890	03711223344	123 Walnut St, City	F	laila@example.com	Mother
9	G29	Mr. Imran	9012345678901	03811223344	456 Redwood St, City	M	imran@example.com	Father
10	G30	Ms. Zara	0123456789012	03911223344	789 Sycamore St, City	F	zara@example.com	Mother

Fee Detail:

Results Messages

	Challan_No	Amount	Discount	Status	Due_Date	Net_Amount
1	F21	15000	2000	Paid	2023-01-10	13000
2	F22	16000	2500	UnPaid	2023-01-12	13500
3	F23	14000	1500	Paid	2023-01-14	12500
4	F24	18000	3000	Paid	2023-01-16	15000
5	F25	17000	2800	UnPaid	2023-01-18	14200
6	F26	15500	2200	UnPaid	2023-01-20	13300
7	F27	16500	2600	Paid	2023-01-22	13900
8	F28	14500	1800	Paid	2023-01-24	12700
9	F29	17500	2700	UnPaid	2023-01-26	14800
10	F30	15800	2300	Paid	2023-01-28	13500

History:

Results Messages

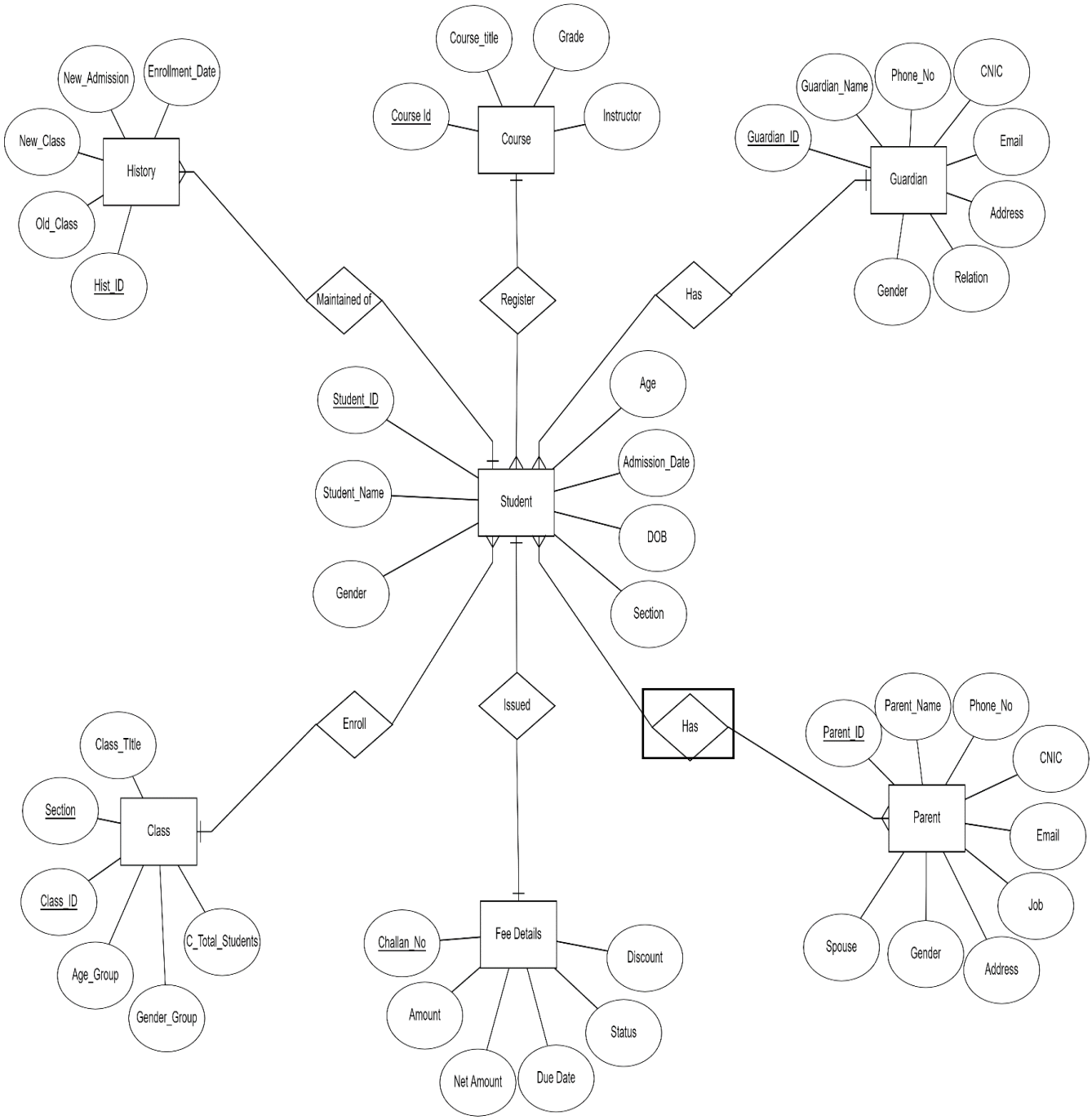
	Hist_ID	Old_Class	New_Class	New_Admission	Enrollment_Date	Student_ID
1	H21	11	12	S22	2023-01-01	S21
2	H22	12	13	S23	2023-01-01	S22
3	H23	13	14	S24	2023-01-01	S23
4	H24	14	15	S25	2023-01-01	S24
5	H25	15	16	S26	2023-01-01	S25
6	H26	16	17	S27	2023-01-01	S26
7	H27	17	18	S28	2023-01-01	S27
8	H28	18	19	S29	2023-01-01	S28
9	H29	19	20	S30	2023-01-01	S29
10	H30	20	11	S21	2023-01-01	S30

Has:

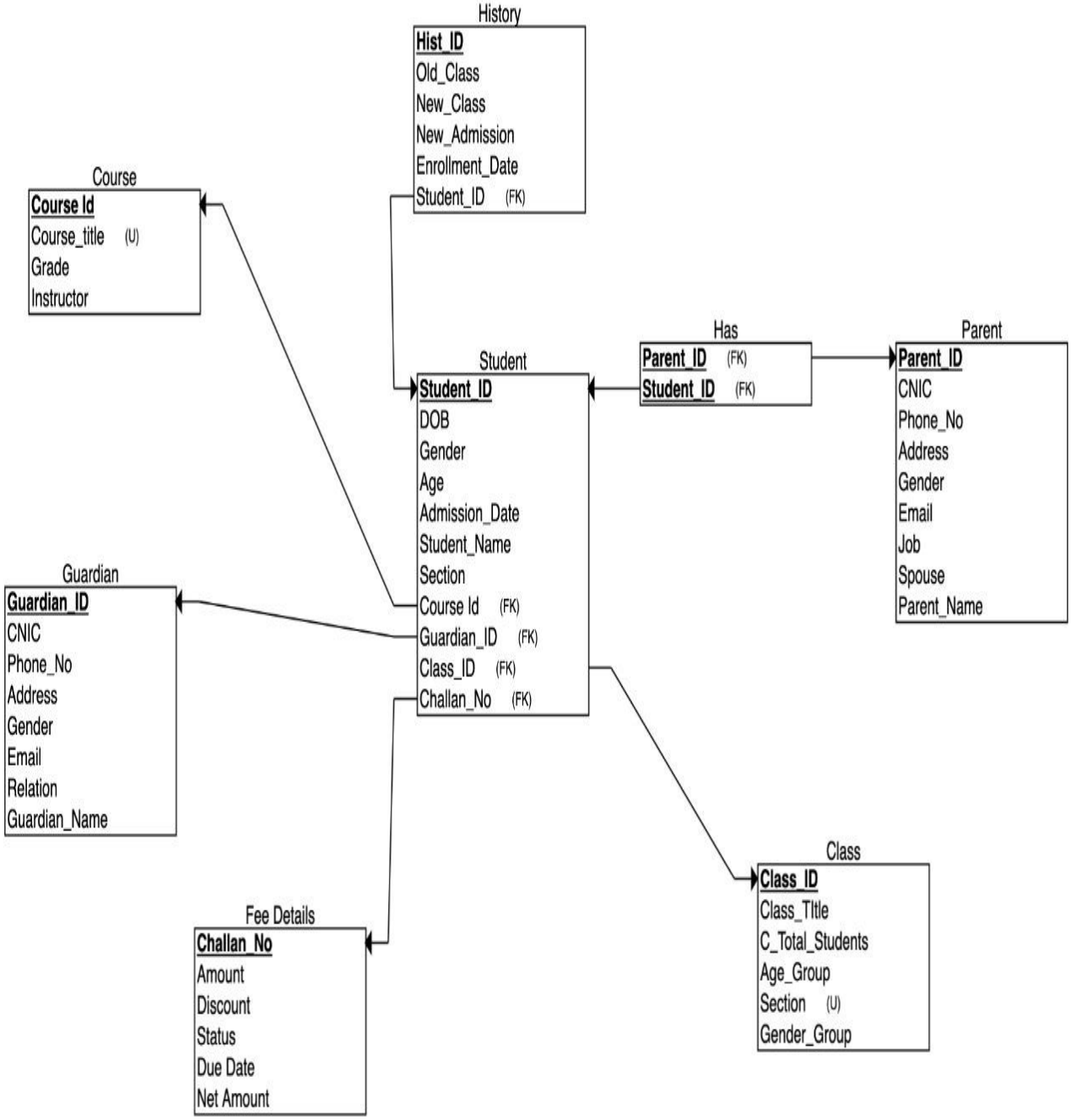
Results Messages

	Parent_ID	Student_ID
1	P22	S22
2	P23	S23
3	P24	S24
4	P25	S25
5	P26	S26
6	P27	S27
7	P28	S28
8	P29	S29
9	P30	S30

2) ERD Diagram using crowfoot notation:



3) Relational schemas:



4) SQL Queries:

1. Make 10 SQL query questions in which the following operators should be used:

a. AND, OR, NOT, IN, NOT IN, BETWEEN, LIKE operator, Distinct,

Group by and Aggregate functions, Insert, Update, Delete.

b. Write SQL answers of 10 SQL query questions.

1)
```
SELECT * FROM STUDENT WHERE Gender = 'M' AND Age > 15 OR Age < 10;
```

100 %

Results | Messages

	Student_ID	Student_Name	DOB	Gender	Age	Admission_Date	Course_Id	Parent_ID	Guardian_ID	Class_ID	Challan_No	Section
1	S21	Ali	2015-03-01	M	9	2023-01-01	C21	P21	G21	11	F21	A
2	S29	Imran	2007-12-15	M	17	2023-01-01	C29	P29	G29	19	F29	I

2)
```
SELECT * FROM PARENT WHERE NOT Gender = 'M';
```

100 %

Results | Messages

	Parent_ID	Parent_Name	CNIC	Phone_No	Address	Gender	Email	P_JOB	Spouse
1	P22	Ms. Ayesha	2345678901234	03111223344	456 Elm St, City	F	ayesha@example.com	Doctor	Kareem
2	P24	Ms. Fatima	4567890123456	03311223344	901 Pine St, City	F	fatima@example.com	Artist	Ahmed
3	P26	Ms. Sara	6789012345678	03511223344	567 Cedar St, City	F	sara@example.com	Pilot	Zaid
4	P28	Ms. Laila	8901234567890	03711223344	123 Walnut St, City	F	laila@example.com	Architect	Ali
5	P30	Ms. Zara	0123456789012	03911223344	789 Sycamore St, City	F	zara@example.com	Doctor	Imran

3)
```
SELECT * FROM COURSE WHERE Course_Id IN ('C21', 'C22', 'C23');
```

100 %

Results | Messages

	Course_Id	Course_title	Grade	Instructor
1	C21	English Literature	A	Mr. Ali
2	C22	Computer Science	A	Ms. Sara
3	C23	Economics	A	Mr. Ahmed

4)
```
SELECT * FROM GUARDIAN WHERE Guardian_ID NOT IN ('G21', 'G22', 'G23');
```

Results | Messages

	Guardian_ID	Guardian_Name	CNIC	Phone_No	Address	Gender	Email	Relation
1	G24	Ms. Fatima	4567890123456	03311223344	901 Pine St, City	F	fatima@example.com	Mother
2	G25	Mr. Zaid	5678901234567	03411223344	234 Maple St, City	M	zaid@example.com	Father
3	G26	Ms. Sara	6789012345678	03511223344	567 Cedar St, City	F	sara@example.com	Mother
4	G27	Mr. Ali	7890123456789	03611223344	890 Birch St, City	M	ali@example.com	Father
5	G28	Ms. Laila	8901234567890	03711223344	123 Walnut St, City	F	laila@example.com	Mother
6	G29	Mr. Imran	9012345678901	03811223344	456 Redwood St, City	M	imran@example.com	Father
7	G30	Ms. Zara	0123456789012	03911223344	789 Sycamore St, City	F	zara@example.com	Mother

5) `SELECT * FROM FEE_DETAILS WHERE Due_Date BETWEEN '2023-01-01' AND '2023-06-30';`

100 %

Results | Messages

	Challan_No	Amount	Discount	Status	Due_Date	Net_Amount
1	F21	15000	2000	Paid	2023-01-10	13000
2	F22	16000	2500	Paid	2023-01-12	13500
3	F23	14000	1500	Paid	2023-01-14	12500
4	F24	18000	3000	Paid	2023-01-16	15000
5	F25	17000	2800	Paid	2023-01-18	14200
6	F26	15500	2200	Paid	2023-01-20	13300
7	F27	16500	2600	Paid	2023-01-22	13900
8	F28	14500	1800	Paid	2023-01-24	12700
9	F29	17500	2700	Paid	2023-01-26	14800
10	F30	15800	2300	Paid	2023-01-28	13500

6) `SELECT * FROM STUDENT WHERE Student_Name LIKE 'A%';`

100 %

Results | Messages

	Student_ID	Student_Name	DOB	Gender	Age	Admission_Date	Course_Id	Parent_ID	Guardian_ID	Class_ID	Challan_No	Section
1	S21	Ali	2015-03-01	M	9	2023-01-01	C21	P21	G21	11	F21	A
2	S23	Ahmed	2013-08-15	M	11	2023-01-01	C23	P23	G23	13	F23	C
3	S26	Ayesha	2010-04-30	F	14	2023-01-01	C26	P26	G26	16	F26	F

7) `SELECT DISTINCT Grade FROM COURSE;`

100 %

Results | Messages

	Grade
1	A

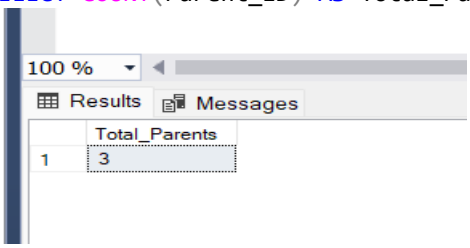
8) `SELECT Class_ID, COUNT(*) AS Total_Students FROM STUDENT GROUP BY Class_ID;`

100 %

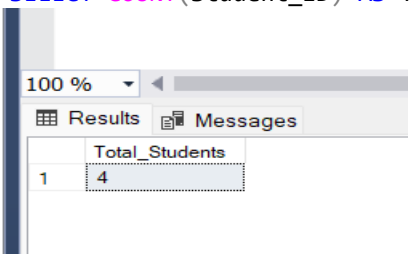
Results | Messages

	Class_ID	Total_Students
1	11	1
2	12	1
3	13	1
4	14	1
5	15	1
6	16	1
7	17	1
8	18	1
9	19	1
10	20	1

9)
```
SELECT COUNT(Parent_ID) AS Total_Parents FROM PARENT WHERE Spouse LIKE 'A%';
```

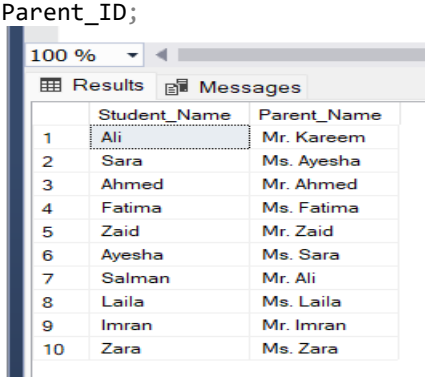


10)
```
SELECT COUNT(Student_ID) AS Total_Students FROM STUDENT WHERE Student_Name LIKE '%m%';
```



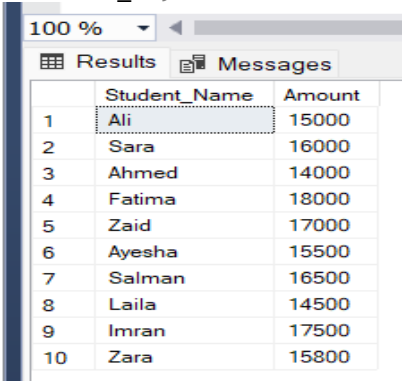
2. Make 10 SQL query questions in which SQL Joins should be used. Write SQL query answers for each SQL Join question.
a. Write SQL query answers for 10 questions.

```
1: INNER JOIN to retrieve student and parent information
SELECT s.Student_Name, p.Parent_Name FROM STUDENT s INNER JOIN PARENT p ON s.Parent_ID=
p.Parent_ID;
```



	Student_Name	Parent_Name
1	Ali	Mr. Kareem
2	Sara	Ms. Ayesha
3	Ahmed	Mr. Ahmed
4	Fatima	Ms. Fatima
5	Zaid	Mr. Zaid
6	Ayesha	Ms. Sara
7	Salman	Mr. Ali
8	Laila	Ms. Laila
9	Imran	Mr. Imran
10	Zara	Ms. Zara

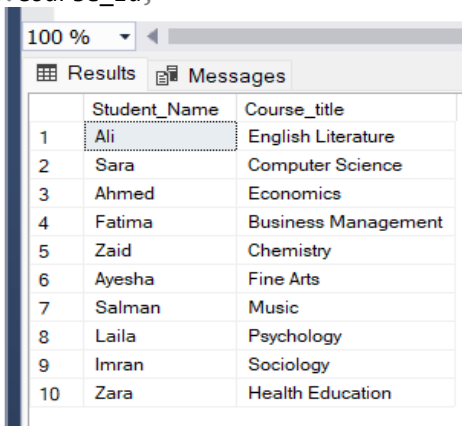
```
2: LEFT JOIN to retrieve all students and their fee details
SELECT s.Student_Name, f.Amount FROM STUDENT s LEFT JOIN FEE_DETAILS f ON s.Challan_No =
f.Challan_No;
```



	Student_Name	Amount
1	Ali	15000
2	Sara	16000
3	Ahmed	14000
4	Fatima	18000
5	Zaid	17000
6	Ayesha	15500
7	Salman	16500
8	Laila	14500
9	Imran	17500
10	Zara	15800

3: INNER JOIN to get the course details of students

```
SELECT s.Student_Name, c.Course_title FROM STUDENT s INNER JOIN COURSE c ON s.Course_Id =
c.Course_Id;
```



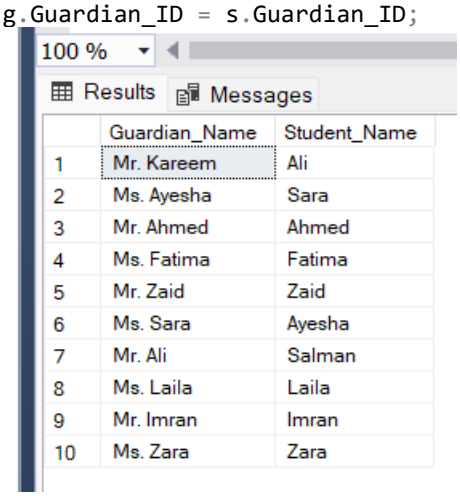
100 %

Results | Messages

	Student_Name	Course_title
1	Ali	English Literature
2	Sara	Computer Science
3	Ahmed	Economics
4	Fatima	Business Management
5	Zaid	Chemistry
6	Ayesha	Fine Arts
7	Salman	Music
8	Laila	Psychology
9	Imran	Sociology
10	Zara	Health Education

4: RIGHT JOIN to find all guardians and their wards

```
SELECT g.Guardian_Name, s.Student_Name FROM GUARDIAN g RIGHT JOIN STUDENT s ON
g.Guardian_ID = s.Guardian_ID;
```



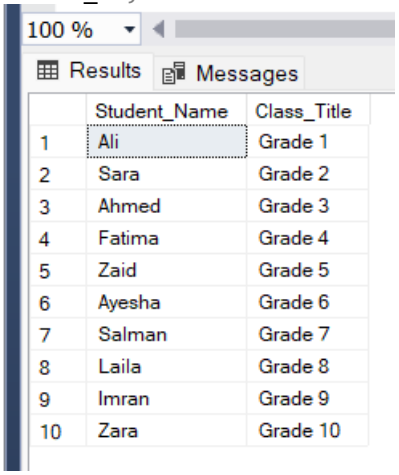
100 %

Results | Messages

	Guardian_Name	Student_Name
1	Mr. Kareem	Ali
2	Ms. Ayesha	Sara
3	Mr. Ahmed	Ahmed
4	Ms. Fatima	Fatima
5	Mr. Zaid	Zaid
6	Ms. Sara	Ayesha
7	Mr. Ali	Salman
8	Ms. Laila	Laila
9	Mr. Imran	Imran
10	Ms. Zara	Zara

5: FULL OUTER JOIN to retrieve all students and their classes

```
SELECT s.Student_Name, c.Class_Title FROM STUDENT s FULL OUTER JOIN CLASS c ON s.Class_ID =
c.Class_ID;
```



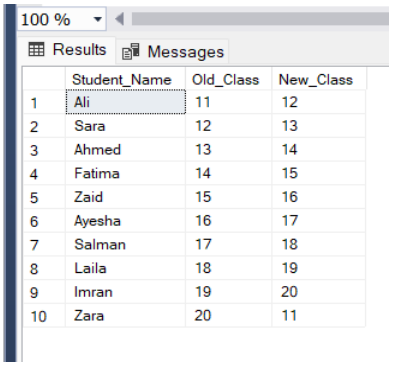
100 %

Results | Messages

	Student_Name	Class_Title
1	Ali	Grade 1
2	Sara	Grade 2
3	Ahmed	Grade 3
4	Fatima	Grade 4
5	Zaid	Grade 5
6	Ayesha	Grade 6
7	Salman	Grade 7
8	Laila	Grade 8
9	Imran	Grade 9
10	Zara	Grade 10

6: LEFT JOIN to get the history of students

```
SELECT s.Student_Name, h.Old_Class, h.New_Class FROM STUDENT s LEFT JOIN HISTORY h ON
s.Student_ID = h.Student_ID;
```



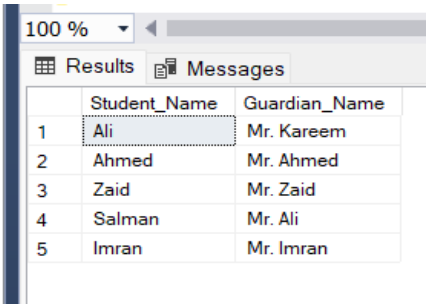
100 %

Results | Messages

	Student_Name	Old_Class	New_Class
1	Ali	11	12
2	Sara	12	13
3	Ahmed	13	14
4	Fatima	14	15
5	Zaid	15	16
6	Ayesha	16	17
7	Salman	17	18
8	Laila	18	19
9	Imran	19	20
10	Zara	20	11

7: INNER JOIN to find students and their guardians who are fathers

```
SELECT s.Student_Name, g.Guardian_Name FROM STUDENT s INNER JOIN GUARDIAN g ON
s.Guardian_ID = g.Guardian_ID WHERE g.Relation = 'Father';
```



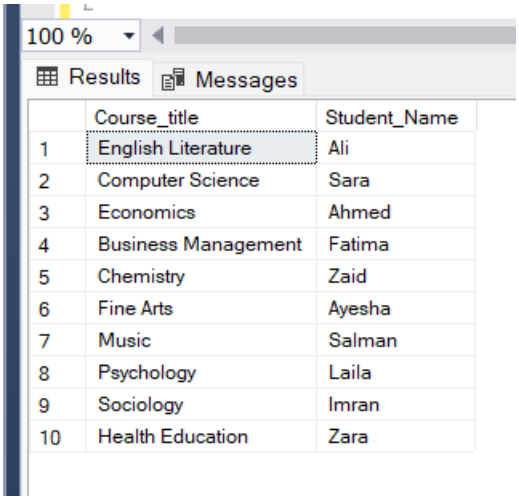
100 %

Results | Messages

	Student_Name	Guardian_Name
1	Ali	Mr. Kareem
2	Ahmed	Mr. Ahmed
3	Zaid	Mr. Zaid
4	Salman	Mr. Ali
5	Imran	Mr. Imran

8: LEFT JOIN to list all courses and the students enrolled in them

```
SELECT c.Course_title, s.Student_Name FROM COURSE c LEFT JOIN STUDENT s ON c.Course_Id =
s.Course_Id;
```



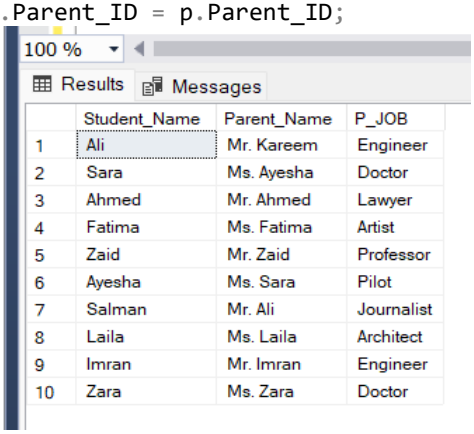
100 %

Results | Messages

	Course_title	Student_Name
1	English Literature	Ali
2	Computer Science	Sara
3	Economics	Ahmed
4	Business Management	Fatima
5	Chemistry	Zaid
6	Fine Arts	Ayesha
7	Music	Salman
8	Psychology	Laila
9	Sociology	Imran
10	Health Education	Zara

9: INNER JOIN to find all students and their parents' jobs

```
SELECT s.Student_Name, p.Parent_Name, p.P_JOB FROM STUDENT s INNER JOIN PARENT p ON
s.Parent_ID = p.Parent_ID;
```



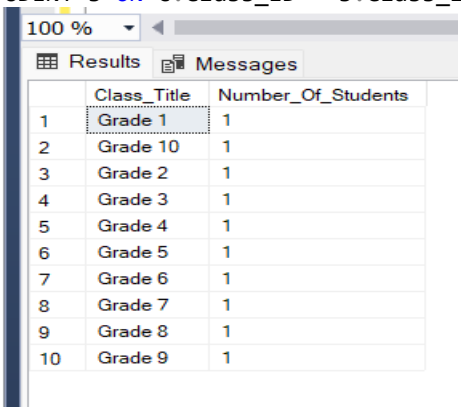
100 %

Results | Messages

	Student_Name	Parent_Name	P_JOB
1	Ali	Mr. Kareem	Engineer
2	Sara	Ms. Ayesha	Doctor
3	Ahmed	Mr. Ahmed	Lawyer
4	Fatima	Ms. Fatima	Artist
5	Zaid	Mr. Zaid	Professor
6	Ayesha	Ms. Sara	Pilot
7	Salman	Mr. Ali	Journalist
8	Laila	Ms. Laila	Architect
9	Imran	Mr. Imran	Engineer
10	Zara	Ms. Zara	Doctor

10: RIGHT JOIN to find all students and the corresponding courses they are enrolled in

```
SELECT c.Class_Title, COUNT(s.Student_ID) AS Number_Of_Students FROM CLASS c LEFT JOIN
STUDENT s ON c.Class_ID = s.Class_ID AND c.Section = s.Section GROUP BY c.Class_Title;
```



100 %

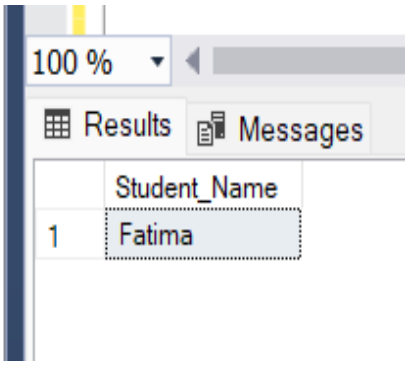
Results | Messages

	Class_Title	Number_Of_Students
1	Grade 1	1
2	Grade 10	1
3	Grade 2	1
4	Grade 3	1
5	Grade 4	1
6	Grade 5	1
7	Grade 6	1
8	Grade 7	1
9	Grade 8	1
10	Grade 9	1

3. Make 10 SQL query questions in which Nested Queries should be used.
Write SQL query answers for each Nested query question.

1: List the students who have enrolled in courses taught by 'Ms. Mahibah'.

```
SELECT Student_Name FROM STUDENT
WHERE Course_Id IN (
    SELECT Course_Id FROM COURSE WHERE Instructor = 'Ms. Mahibah'
);
```



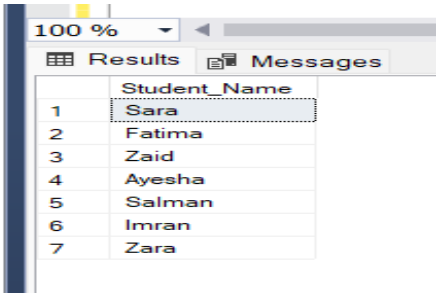
100 %

Results | Messages

	Student_Name
1	Fatima

2: Find the names of students who have a discount greater than 2000 on their fee.

```
SELECT Student_Name FROM STUDENT
WHERE Challan_No IN (
    SELECT Challan_No FROM FEE_DETAILS WHERE CAST(Discount AS INT) > 2000
);
```



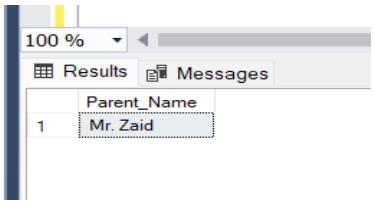
100 %

Results | Messages

	Student_Name
1	Sara
2	Fatima
3	Zaid
4	Ayesha
5	Salman
6	Imran
7	Zara

3: Find the parents who have children enrolled in 'Grade 5'.

```
SELECT Parent_Name FROM PARENT
WHERE Parent_ID IN (
    SELECT Parent_ID FROM STUDENT WHERE Class_ID = '15'
);
```



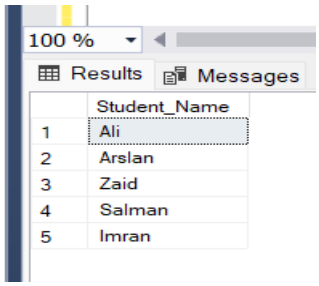
100 %

Results | Messages

	Parent_Name
1	Mr. Zaid

4 : Find students whose guardians' gender is male.

```
SELECT Student_Name FROM STUDENT
WHERE Guardian_ID IN (
    SELECT Guardian_ID FROM GUARDIAN WHERE Gender='M'
);
```



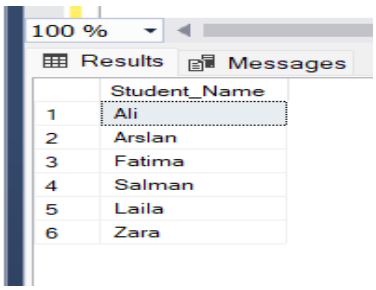
100 %

Results | Messages

	Student_Name
1	Ali
2	Arslan
3	Zaid
4	Salman
5	Imran

5: List the names of students whose fee status is 'Paid'.

```
SELECT Student_Name FROM STUDENT
WHERE Challan_No IN (
    SELECT Challan_No FROM FEE_DETAILS WHERE Status = 'Paid'
);
```



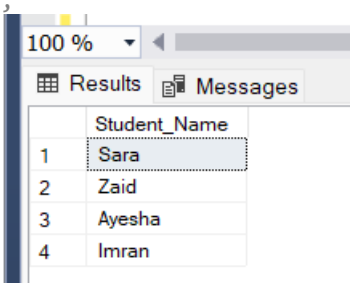
100 %

Results | Messages

	Student_Name
1	Ali
2	Arslan
3	Fatima
4	Salman
5	Laila
6	Zara

6: List the names of students whose fee status is 'UnPaid'.

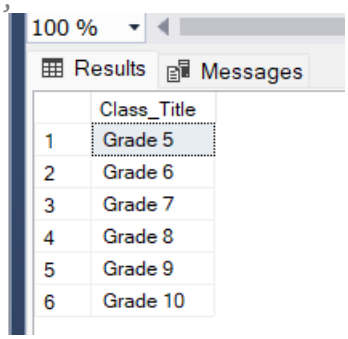
```
SELECT Student_Name FROM STUDENT
WHERE Challan_No IN (
    SELECT Challan_No FROM FEE_DETAILS WHERE Status = 'UnPaid'
);
```



	Student_Name
1	Sara
2	Zaid
3	Ayesha
4	Imran

7: Find the classes with total students greater than 25.

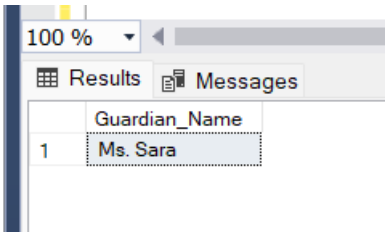
```
SELECT Class_Title FROM CLASS
   WHERE Class_ID IN (
    SELECT Class_ID FROM CLASS WHERE CAST(C_Total_Students AS INT) > 25
);
```



	Class_Title
1	Grade 5
2	Grade 6
3	Grade 7
4	Grade 8
5	Grade 9
6	Grade 10

8: Find the guardians who are 'Mother' and have children in 'Grade 6'.

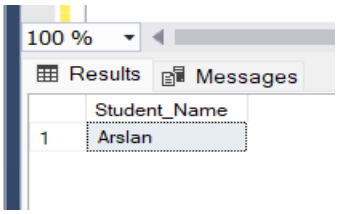
```
SELECT Guardian_Name FROM GUARDIAN
WHERE Guardian_ID IN (
    SELECT Guardian_ID FROM STUDENT WHERE Class_ID = '16'
) AND Relation = 'Mother';
```



	Guardian_Name
1	Ms. Sara

9: Using EXISTS to find students enrolled in class 13.

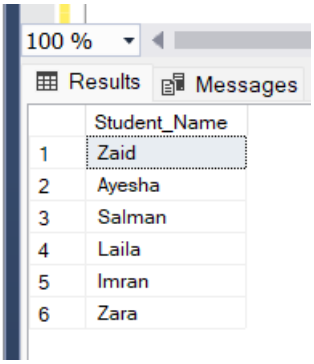
```
SELECT Student_Name FROM STUDENT s
WHERE EXISTS (
    SELECT 1
    FROM CLASS c WHERE c.Class_ID = s.Class_ID AND c.Class_ID = '13'
);
```



	Student_Name
1	Arslan

10: Find the names of students who are enrolled in classes where the average age of students is greater than 12.

```
SELECT Student_Name FROM STUDENT
WHERE Class_ID IN (
    SELECT Class_ID
    FROM STUDENT GROUP BY Class_ID HAVING AVG(Age) > 12
);
```



100 %

Results | Messages

	Student_Name
1	Zaid
2	Ayesha
3	Salman
4	Laila
5	Imran
6	Zara

Bonus Marks

Make 15SQL query questions in which the following topics should be used

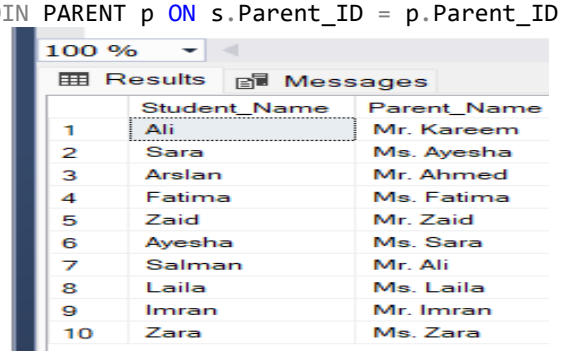
1. Views

2. Stored Procedures

3. Triggers

1: Create a view that lists all students along with their parent's names.

```
CREATE VIEW StudentParentView AS
SELECT s.Student_Name, p.Parent_Name FROM STUDENT s
JOIN PARENT p ON s.Parent_ID = p.Parent_ID;
```



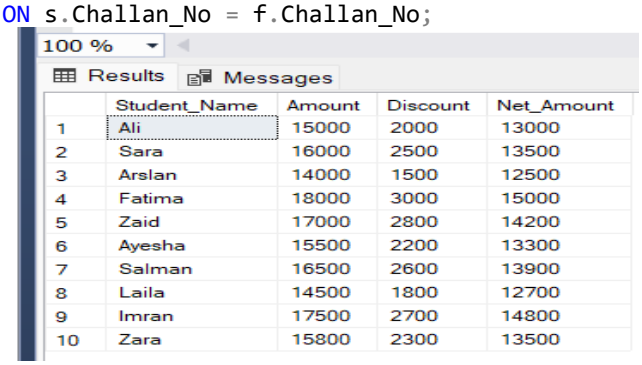
100 %

Results | Messages

	Student_Name	Parent_Name
1	Ali	Mr. Kareem
2	Sara	Ms. Ayesha
3	Arslan	Mr. Ahmed
4	Fatima	Ms. Fatima
5	Zaid	Mr. Zaid
6	Ayesha	Ms. Sara
7	Salman	Mr. Ali
8	Laila	Ms. Laila
9	Imran	Mr. Imran
10	Zara	Ms. Zara

2: Create a view that shows students along with their fee details (amount, discount, net amount).

```
CREATE VIEW StudentFeeView AS
SELECT s.Student_Name, f.Amount, f.Discount, f.Net_Amount FROM STUDENT s JOIN FEE_DETAILS f
ON s.Challan_No = f.Challan_No;
```



100 %

Results | Messages

	Student_Name	Amount	Discount	Net_Amount
1	Ali	15000	2000	13000
2	Sara	16000	2500	13500
3	Arslan	14000	1500	12500
4	Fatima	18000	3000	15000
5	Zaid	17000	2800	14200
6	Ayesha	15500	2200	13300
7	Salman	16500	2600	13900
8	Laila	14500	1800	12700
9	Imran	17500	2700	14800
10	Zara	15800	2300	13500

3: Create a view that lists the courses along with the number of students enrolled in each course.

```
CREATE VIEW CourseStudentCountView AS
SELECT c.Course_title, COUNT(s.Student_ID) AS Student_Count FROM COURSE c
JOIN STUDENT s ON c.Course_Id = s.Course_Id GROUP BY c.Course_title;
```

Results Messages

	Course_title	Student_Count
1	Business Management	1
2	Chemistry	1
3	Computer Science	1
4	Economics	1
5	English Literature	1
6	Fine Arts	1
7	Health Education	1
8	Music	1
9	Psychology	1
10	Sociology	1

4: Create a view that shows the history of students who changed classes.

```
CREATE VIEW StudentHistoryView AS
SELECT s.Student_Name, h.Old_Class, h.New_Class, h.Enrollment_Date
FROM STUDENT s JOIN HISTORY h ON s.Student_ID = h.Student_ID;
```

Results Messages

	Student_Name	Old_Class	New_Class	Enrollment_Date
1	Ali	11	12	2023-01-01
2	Sara	12	13	2023-01-01
3	Arslan	13	14	2023-01-01
4	Fatima	14	15	2023-01-01
5	Zaid	15	16	2023-01-01
6	Ayesha	16	17	2023-01-01
7	Salman	17	18	2023-01-01
8	Laila	18	19	2023-01-01
9	Imran	19	20	2023-01-01
10	Zara	20	11	2023-01-01

5: Create a view that lists all guardians along with the names of the students they are responsible for.

```
CREATE VIEW GuardianStudentView AS
SELECT g.Guardian_Name, s.Student_Name
FROM GUARDIAN g JOIN STUDENT s ON g.Guardian_ID = s.Guardian_ID;
```

Results Messages

	Guardian_Name	Student_Name
1	Mr. Kareem	Ali
2	Ms. Ayesha	Sara
3	Mr. Ahmed	Arslan
4	Ms. Fatima	Fatima
5	Mr. Zaid	Zaid
6	Ms. Sara	Ayesha
7	Mr. Ali	Salman
8	Ms. Laila	Laila
9	Mr. Imran	Imran
10	Ms. Zara	Zara

6: This procedure retrieves details of a student from the STUDENT table based on Student_ID.

```
CREATE OR ALTER PROCEDURE GetStudentDetails
    @Student_ID VARCHAR(255)
AS
BEGIN
    SELECT *
    FROM STUDENT
    WHERE Student_ID = @Student_ID;
END;

EXEC GetStudentDetails @Student_ID = 'S23';
```

Results Messages

	Student_ID	Student_Name	DOB	Gender	Age	Admission_Date	Course_Id	Parent_ID	Guardian_ID	Class_ID	Challan_No	Section
1	S23	Arslan	2013-08-15	M	11	2023-01-01	C23	P23	G23	13	F23	C

7: This procedure retrieves enrollment history (changes in classes) for a student from the HISTORY table.

```
CREATE OR ALTER PROCEDURE GetClassEnrollmentHistory
    @Student_ID VARCHAR(255)
AS
BEGIN
    SELECT *
    FROM HISTORY
    WHERE Student_ID = @Student_ID;
END;

EXEC GetClassEnrollmentHistory @Student_ID = 'S23';
```

133 % Results Messages

	Hist_ID	Old_Class	New_Class	New_Admission	Enrollment_Date	Student_ID
1	H23	13	14	S24	2023-01-01	S23

8: This procedure retrieves guardian information based on the provided student ID.

```
CREATE PROCEDURE GetGuardianInfoByStudentID
    @student_id VARCHAR(255)
AS
BEGIN
    SELECT G.*
    FROM GUARDIAN G
    JOIN STUDENT S ON G.Guardian_ID = S.Guardian_ID
    WHERE S.Student_ID = @student_id;
END;

EXEC GetGuardianInfoByStudentID @Student_ID = 'S23';
```

133 % Results Messages

	Guardian_ID	Guardian_Name	CNIC	Phone_No	Address	Gender	Email	Relation
1	G23	Mr. Ahmed	3456789012345	03211223344	789 Oak St, City	M	ahmed@example.com	Father

9: This trigger automatically inserts a corresponding record into the HAS table whenever a new student is inserted into the STUDENT table.

```
CREATE TRIGGER trgAfterInsertStudent
ON STUDENT
AFTER INSERT
AS
BEGIN
    INSERT INTO HAS (Parent_ID, Student_ID)
    SELECT i.Parent_ID, i.Student_ID
    FROM inserted i;
END;
```

10: This trigger updates the corresponding record in the HAS table whenever a student's parent ID is updated in the STUDENT table.

```
CREATE TRIGGER trgAfterUpdateStudentParent
ON STUDENT
AFTER UPDATE
AS
BEGIN
    IF UPDATE(Parent_ID)
    BEGIN
        UPDATE H
        SET H.Parent_ID = i.Parent_ID
        FROM HAS H
        JOIN inserted i ON H.Student_ID = i.Student_ID;
    END;
END;
```