Republic of the Philippines

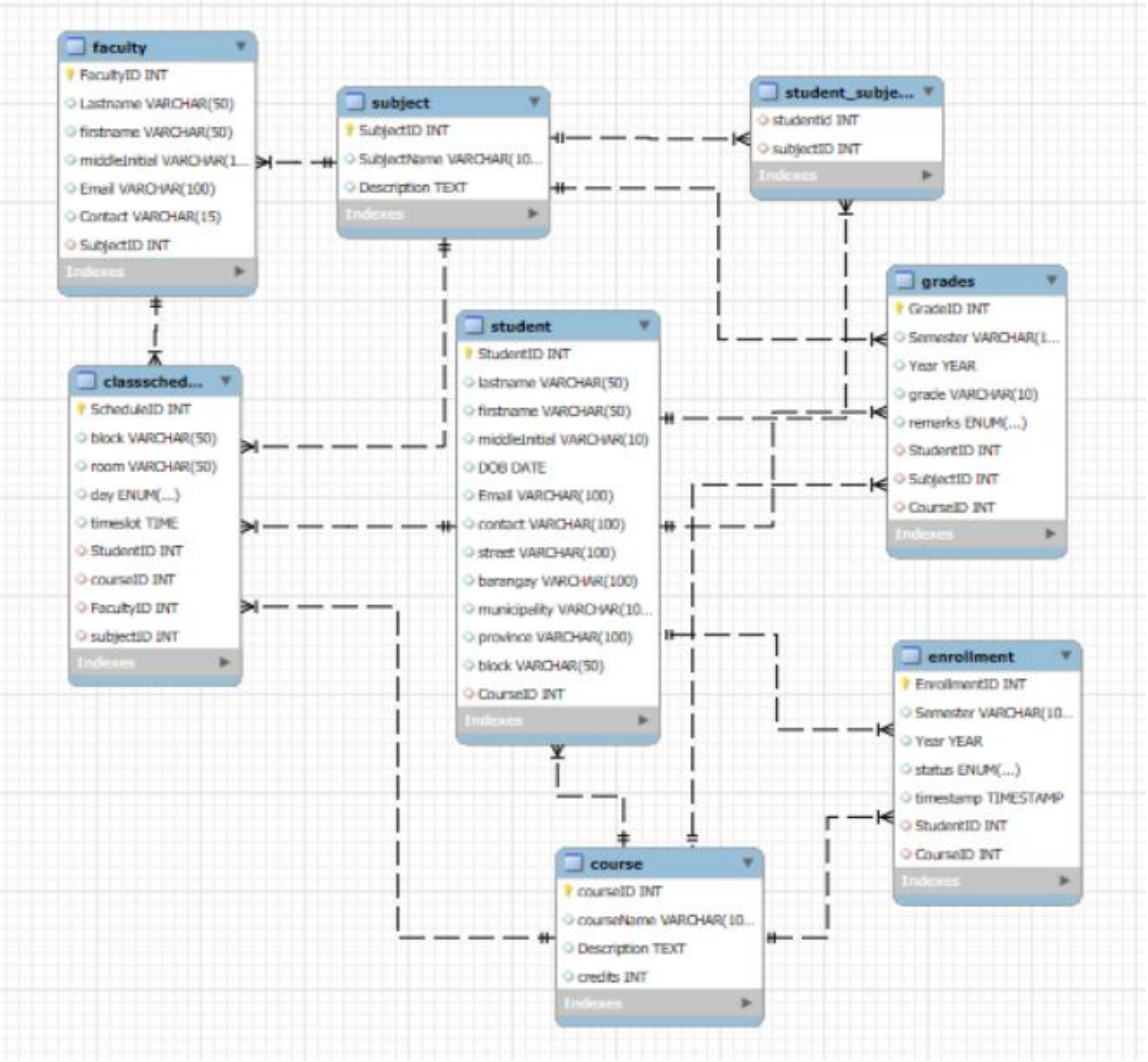
BICOL UNIVERSITY POLANGUI

Polangui, Albay

Registrar Student Management Systems for Computer Studies Department of Bicol University Polangui

Our group proposes the development of a Registrar Management System specifically designed for the Computer Studies Department of Bicol University Polangui. This system aims to centralize and streamline the management of student-related information, class schedules, and personal records. By digitizing these processes, the system will help improve data organization, enhance communication between students and faculty, and support better decision-making within the department. The ultimate goal is to create a more efficient and transparent environment for academic management and student support.

Final ER Diagram



Database schema and table descriptions:

RSMS Database – Table Summary

Table: course

Column	Data Type	Description
courseID	INT	Primary Key, Auto Increment
courseName	VARCHAR(100)	Name of the course
Description	TEXT	Course description
credits	INT	Number of credits

Table: subject

Column	Data Type	Description
SubjectID	INT	Primary Key, Auto Increment
SubjectName	VARCHAR(100)	Name of the subject
Description	TEXT	Subject description

Table: student

Column	Data Type	Description
StudentID	INT	Primary Key, Auto Increment
lastname	VARCHAR(50)	Student's last name
firstname	VARCHAR(50)	Student's first name
middleInitial	VARCHAR(10)	Student's middle initial
DOB	DATE	Date of birth
Email	VARCHAR(100)	Email address
contact	VARCHAR(100)	Contact number
street	VARCHAR(100)	Street address
barangay	VARCHAR(100)	Barangay
municipality	VARCHAR(100)	Municipality
province	VARCHAR(100)	Province
block	VARCHAR(50)	Block name or section
CourseID	INT	FK → course(courseID)

Table: faculty

Column	Data Type	Description
FacultyID	INT	Primary Key, Auto Increment
lastname	VARCHAR(50)	Last name
firstname	VARCHAR(50)	First name
middleInitial	VARCHAR(10)	Middle initial
Email	VARCHAR(100)	Email address
Contact	VARCHAR(15)	Contact number
SubjectID	INT	FK → subject(SubjectID)

Table: grades

Column	Data Type	Description
GradeID	INT	Primary Key, Auto Increment
Semester	VARCHAR(10)	Semester (e.g., 1st, 2nd)
Year	YEAR	Academic year
grade	VARCHAR(10)	Grade value (e.g., 1.0, 3.0, INC)
remarks	ENUM	('Passed', 'Failed', 'Incomplete')
StudentID	INT	FK → student(StudentID)
SubjectID	INT	FK → subject(SubjectID)
CourseID	INT	FK → course(CourseID)

Table: enrollment

Column	Data Type	Description
EnrollmentID	INT	Primary Key, Auto Increment
Semester	VARCHAR(100)	Semester (e.g., 1st, Summer)
Year	YEAR	Academic year
status	ENUM	('Enrolled', 'Completed', 'Dropped')
timestamp	TIMESTAMP	Default: current timestamp
StudentID	INT	FK → student(StudentID)
CourseID	INT	FK → course(CourseID)

Table: classschedule

Column	Data Type	Description
ScheduleID	INT	Primary Key, Auto Increment
block	VARCHAR(50)	Section/block name
room	VARCHAR(50)	Room number
day	ENUM	('Monday' to 'Sunday')
timeslot	TIME	Class start time
StudentID	INT	FK → student(StudentID)
CourseID	INT	FK → course(CourseID)
FacultyID	INT	FK → faculty(FacultyID)
SubjectID	INT	FK → subject(SubjectID)

Table: student_subject

Column	Data Type	Description
StudentID	INT	FK → student(StudentID) (composite PK)
SubjectID	INT	FK → subject(SubjectID) (composite PK)

SQL queries (basic, BI, optimized)

- Total Credits Earned by Each Student (SUM)

```
SELECT
  s.StudentID,
  CONCAT(s.firstname, ' ', s.lastname) AS StudentName,
  SUM(c.credits) AS TotalCredits
FROM
  student s
JOIN
  course c ON s.CourseID = c.CourseID
JOIN
```

```
    enrollment e ON s.StudentID = e.StudentID
WHERE
    e.status = 'Completed'
GROUP BY
    s.StudentID
HAVING
    SUM(c.credits) > 0
ORDER BY
    TotalCredits DESC;
```

- Number of Students Enrolled Per Course (COUNT)

```
SELECT
    c.courseName,
    COUNT(e.StudentID) AS TotalStudents
FROM
    course c
JOIN
    enrollment e ON c.CourseID = e.CourseID
GROUP BY
    c.CourseID
ORDER BY
    TotalStudents DESC;
```

- Average Grade Per Subject (AVG)

```
SELECT
    sub.SubjectName,
    AVG(CASE WHEN g.remarks = 'Passed' THEN 1 ELSE 0 END) AS AvgGrade
FROM
```

```
    grades g
JOIN
    subject sub ON g.SubjectID = sub.SubjectID
GROUP BY
    sub.SubjectID
HAVING
    COUNT(g.GradeID) > 5 -- Only subjects with more than 5 grades
ORDER BY
    AvgGrade DESC;
```

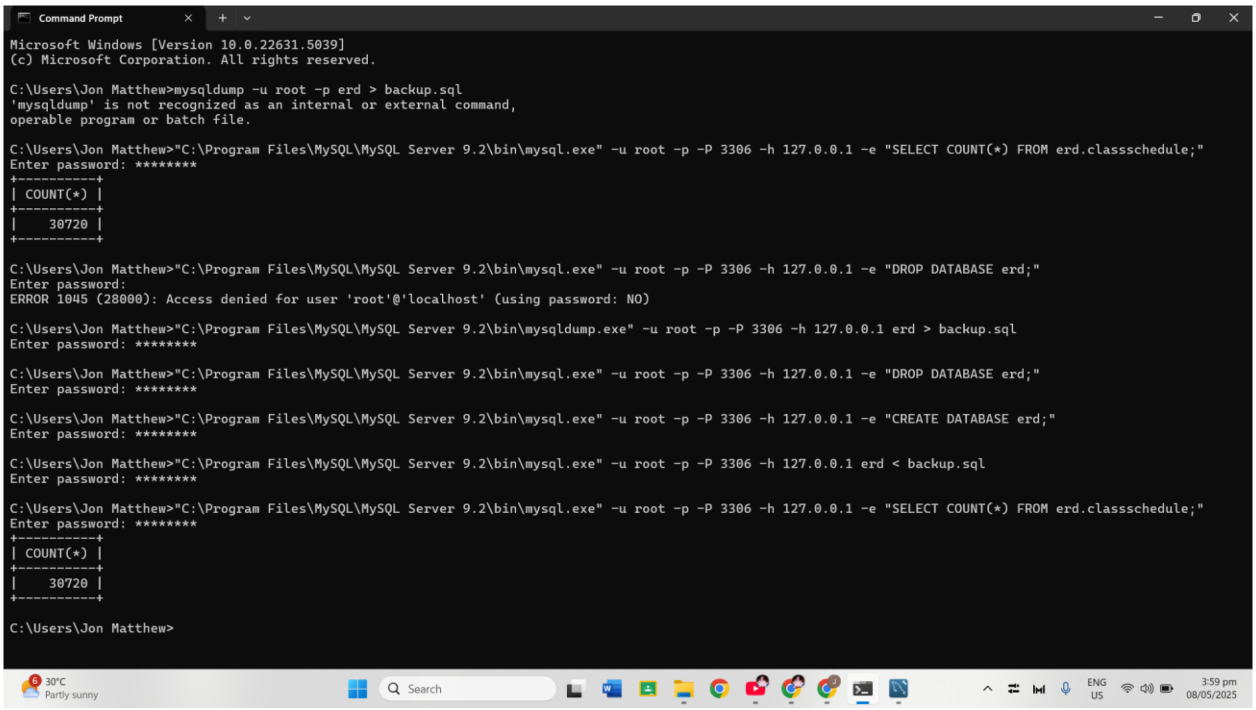
- Faculty Members with Most Enrolled Students (JOIN + COUNT)

```
SELECT
    f.FacultyID,
    CONCAT(f.firstname, ' ', f.lastname) AS FacultyName,
    COUNT(e.StudentID) AS TotalStudents
FROM
    faculty f
JOIN
    subject sub ON f.SubjectID = sub.SubjectID
JOIN
    classschedule cs ON sub.SubjectID = cs.subjectID
JOIN
    enrollment e ON cs.CourseID = e.CourseID AND e.StudentID = cs.StudentID
GROUP BY
    f.FacultyID
ORDER BY
    TotalStudents DESC
LIMIT 12;
```

- Students Who Failed More Than One Subject

```
SELECT
    s.StudentID,
    CONCAT(s.firstname, ' ', s.lastname) AS FullName,
    COUNT(*) AS FailedSubjects
FROM
    grades g
JOIN
    student s ON g.StudentID = s.StudentID
WHERE
    g.remarks = 'Failed'
GROUP BY
    s.StudentID
HAVING
    COUNT(*) > 1;
```

Backup and recovery steps



```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Jon Matthew>mysqldump -u root -p erd > backup.sql
'mysqldump' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Jon Matthew>"C:\Program Files\MySQL\MySQL Server 9.2\bin\mysql.exe" -u root -p -P 3306 -h 127.0.0.1 -e "SELECT COUNT(*) FROM erd.classschedule;"
Enter password: ********
+----------+
| COUNT(*) |
+----------+
|    30720 |
+----------+

C:\Users\Jon Matthew>"C:\Program Files\MySQL\MySQL Server 9.2\bin\mysql.exe" -u root -p -P 3306 -h 127.0.0.1 -e "DROP DATABASE erd;"
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: NO)

C:\Users\Jon Matthew>"C:\Program Files\MySQL\MySQL Server 9.2\bin\mysqldump.exe" -u root -p -P 3306 -h 127.0.0.1 erd > backup.sql
Enter password: ********

C:\Users\Jon Matthew>"C:\Program Files\MySQL\MySQL Server 9.2\bin\mysql.exe" -u root -p -P 3306 -h 127.0.0.1 -e "DROP DATABASE erd;"
Enter password: ********

C:\Users\Jon Matthew>"C:\Program Files\MySQL\MySQL Server 9.2\bin\mysql.exe" -u root -p -P 3306 -h 127.0.0.1 -e "CREATE DATABASE erd;"
Enter password: ********

C:\Users\Jon Matthew>"C:\Program Files\MySQL\MySQL Server 9.2\bin\mysql.exe" -u root -p -P 3306 -h 127.0.0.1 erd < backup.sql
Enter password: ********

C:\Users\Jon Matthew>"C:\Program Files\MySQL\MySQL Server 9.2\bin\mysql.exe" -u root -p -P 3306 -h 127.0.0.1 -e "SELECT COUNT(*) FROM erd.classschedule;"
Enter password: ********
+----------+
| COUNT(*) |
+----------+
|    30720 |
+----------+

C:\Users\Jon Matthew>
```

Indexing and performance improvements

Query 1: Retrieve names and grades of students who failed.

```
EXPLAIN SELECT s.firstname, s.lastname, g.grade, g.remarks

FROM student s

JOIN grades g ON s.StudentID = g.StudentID

WHERE g.remarks = 'Failed';

-- Indexing

CREATE INDEX idx_remarks ON grades(remarks);

-- Optimized Query

EXPLAIN SELECT s.firstname, s.lastname, g.grade, g.remarks

FROM student s

JOIN grades g ON s.StudentID = g.StudentID

WHERE g.remarks = 'Failed';
```

Query 2: List students who completed courses and calculate their total earned credits.

```
EXPLAIN SELECT

    s.StudentID,

    CONCAT(s.firstname, ' ', s.lastname) AS StudentName,

    SUM(c.credits) AS TotalCredits

FROM

    student s

JOIN

    course c ON s.CourseID = c.CourseID

JOIN

    enrollment e ON s.StudentID = e.StudentID

WHERE

    e.status = 'Completed'

GROUP BY

    s.StudentID
```

```
HAVING
    SUM(c.credits) > 0
ORDER BY
    TotalCredits DESC;
-- Indexing
CREATE INDEX idx_student_courseID ON student(CourseID);
CREATE INDEX idx_enrollment_studentID ON enrollment(StudentID);
CREATE INDEX idx_enrollment_status ON enrollment(status);
-- Optimized Query
EXPLAIN SELECT
    s.StudentID,
    CONCAT(s.firstname, ' ', s.lastname) AS StudentName,
    SUM(c.credits) AS TotalCredits
FROM
    student s
JOIN
    course c ON s.CourseID = c.CourseID
JOIN
    enrollment e ON s.StudentID = e.StudentID
WHERE
    e.status = 'Completed'
GROUP BY
    s.StudentID
HAVING
    SUM(c.credits) > 0
ORDER BY
    TotalCredits DESC;
```

Query 3: Get student names, course names, and enrollment timestamps between two dates.

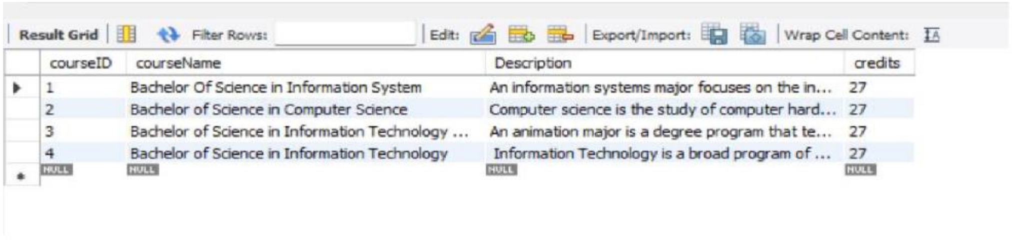
```
EXPLAIN SELECT s.firstname, s.lastname, c.courseName, e.timestamp

FROM student s

JOIN enrollment e ON s.StudentID = e.StudentID

JOIN course c ON e.CourseID = c.CourseID

WHERE e.timestamp BETWEEN '2022-01-01' AND '2023-12-31';

-- Indexing

CREATE INDEX idx_enrollment_timestamp ON enrollment(timestamp);

-- Optimized Query

EXPLAIN SELECT s.firstname, s.lastname, c.courseName, e.timestamp

FROM student s

JOIN enrollment e ON s.StudentID = e.StudentID

JOIN course c ON e.CourseID = c.CourseID

WHERE e.timestamp BETWEEN '2022-01-01' AND '2023-12-31';
```

Role assignments and contribution summary

Role Assignments — Zaijan M. Alvarado: Project Lead Jon Matthew B. Mella : Database Architect Jaiden Nykluz Fermante : SQL Developer Otelo P. Nobleza III : SQL Developer Andrei Lloyd V. Sinfuego : QA Tester Symon Cristoffer B. Cano : QA Tester Contribution Summary by Role — Project Lead (Zaijan M. Alvarado) • Phase 1: Define project scope; design ER diagram; assign roles; set up GitHub (establishes vision & schema) • Phase 6: Oversee final report, README, versioning, and presentation prep (project completion & presentation-ready) Database Architect (Jon Matthew B. Mella) • Phase 3: Develop BI queries; design simplified star schema; produce data-insights report (enables analysis & reporting) SQL Developers (Jaiden Nykluz Fermante & Otelo P. Nobleza III) • Phase 2: Implement physical MySQL schema and seed 100K+ rows across normalized tables (implements core database) • Phase 4: Generate SQL backup file; deploy cloud-hosted schema; run backup-validation scripts (introduces resilience & remote hosting) QA Testers (Andrei Lloyd V. Sinfuego & Symon Cristoffer B. Cano) • Phase 5: Execute indexed-query benchmarks; review performance reports; audit access-control logs (ensures performance & security readiness) • Phase 6: Assist with documentation polishing and final system checks (presentation-readiness)

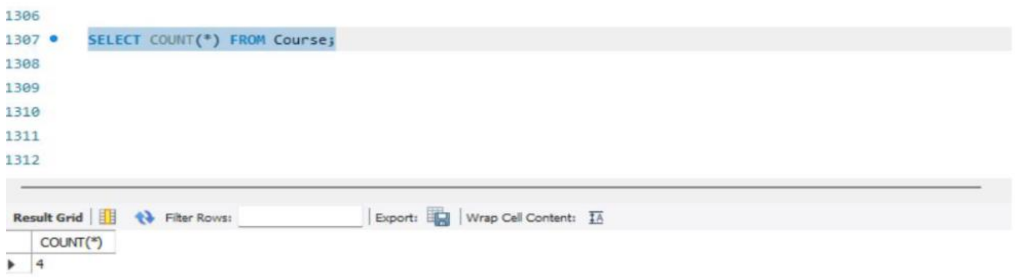
Screenshot

Course:



courseID	courseName	Description	credits
1	Bachelor Of Science in Information System	An information systems major focuses on the in...	27
2	Bachelor of Science in Computer Science	Computer science is the study of computer hard...	27
3	Bachelor of Science in Information Technology ...	An animation major is a degree program that te...	27
4	Bachelor of Science in Information Technology	Information Technology is a broad program of ...	27
NULL	NULL	NULL	NULL

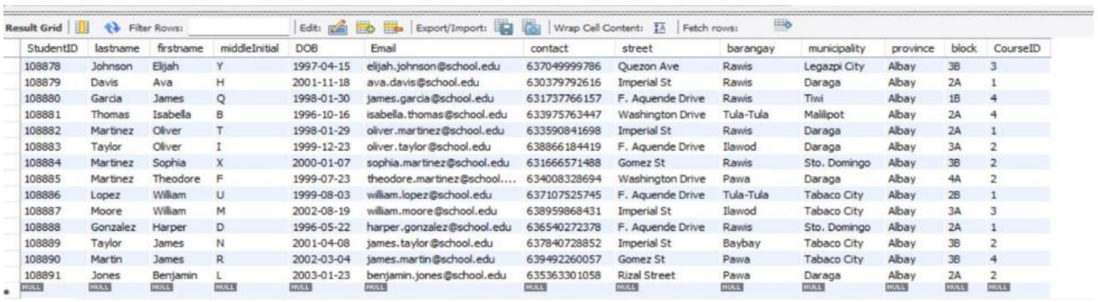
Row Count Per Table:



```
SELECT COUNT(*) FROM Course;
```

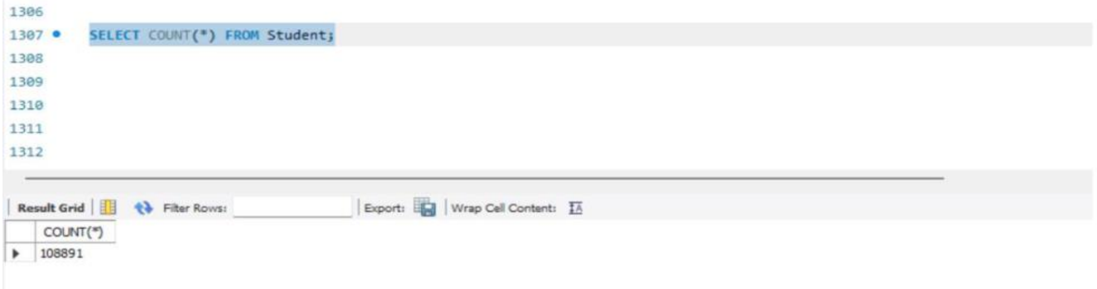
COUNT(*)
4

Student:



StudentID	lastname	firstname	middleInitial	DOB	Email	contact	street	barangay	municipality	province	block	CourseID
108878	Johnson	Elijah	Y	1997-04-15	elijah.johnson@school.edu	637049999786	Quezon Ave	Rawis	Legazpi City	Albay	3B	3
108879	Davis	Ava	H	2001-11-18	ava.davis@school.edu	630379792616	Imperial St	Rawis	Daraga	Albay	2A	1
108880	Garcia	James	Q	1998-01-30	james.garcia@school.edu	631737766157	F. Aquende Drive	Rawis	Tiwi	Albay	1B	4
108881	Thomas	Isabella	B	1996-10-16	isabella.thomas@school.edu	633975763447	Washington Drive	Tula-Tula	Malilipot	Albay	2A	4
108882	Martinez	Oliver	T	1998-01-29	oliver.martinez@school.edu	633590841698	Imperial St	Rawis	Daraga	Albay	2A	1
108883	Taylor	Oliver	I	1999-12-23	oliver.taylor@school.edu	638866184419	F. Aquende Drive	Ilawod	Daraga	Albay	3A	2
108884	Martinez	Sophia	X	2000-01-07	sophia.martinez@school.edu	631666571488	Gomez St	Rawis	Sto. Domingo	Albay	3B	2
108885	Martinez	Theodore	F	1999-07-23	theodore.martinez@school....	634008328694	Washington Drive	Pawa	Daraga	Albay	4A	2
108886	Lopez	William	U	1999-08-03	william.lopez@school.edu	637107525745	F. Aquende Drive	Tula-Tula	Tabaco City	Albay	2B	1
108887	Moore	William	M	2002-08-19	william.moore@school.edu	638959868431	Imperial St	Ilawod	Tabaco City	Albay	3A	3
108888	Gonzalez	Harper	D	1996-05-22	harper.gonzalez@school.edu	636540272378	F. Aquende Drive	Rawis	Sto. Domingo	Albay	2A	1
108889	Taylor	James	N	2001-04-08	james.taylor@school.edu	637840728852	Imperial St	Baybay	Tabaco City	Albay	3B	2
108890	Martin	James	R	2002-03-04	james.martin@school.edu	639492260057	Gomez St	Pawa	Tabaco City	Albay	3B	4
108891	Jones	Benjamin	L	2003-01-23	benjamin.jones@school.edu	635363301058	Rizal Street	Pawa	Daraga	Albay	2A	2
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

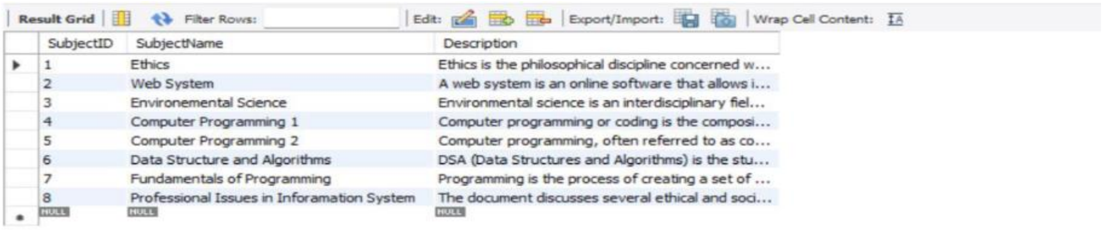
Row Count Per Table:



```
SELECT COUNT(*) FROM Student;
```

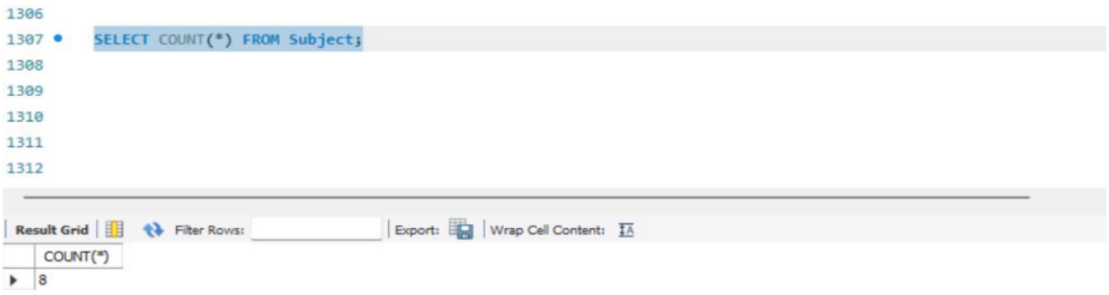
COUNT(*)
108891

Subject:



SubjectID	SubjectName	Description
1	Ethics	Ethics is the philosophical discipline concerned w...
2	Web System	A web system is an online software that allows i...
3	Environemental Science	Environmental science is an interdisciplinary fiel...
4	Computer Programming 1	Computer programming or coding is the composi...
5	Computer Programming 2	Computer programming, often referred to as co...
6	Data Structure and Algorithms	DSA (Data Structures and Algorithms) is the stu...
7	Fundamentals of Programming	Programming is the process of creating a set of ...
8	Professional Issues in Inforamation System	The document discusses several ethical and soci...
NULL	NULL	NULL

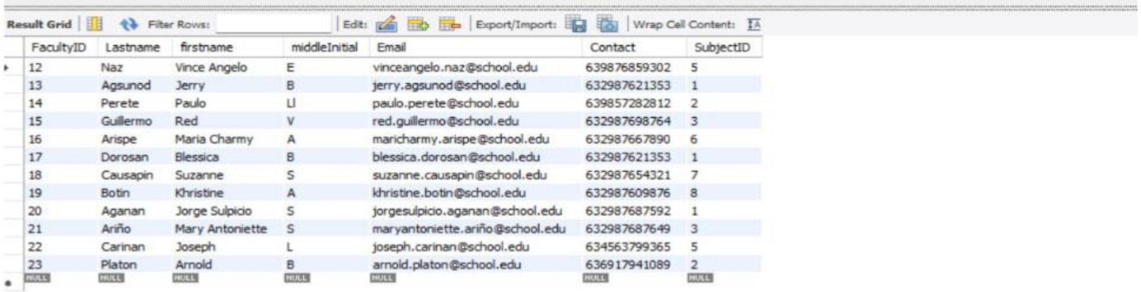
Row Count Per Table:



```
SELECT COUNT(*) FROM Subject;
```

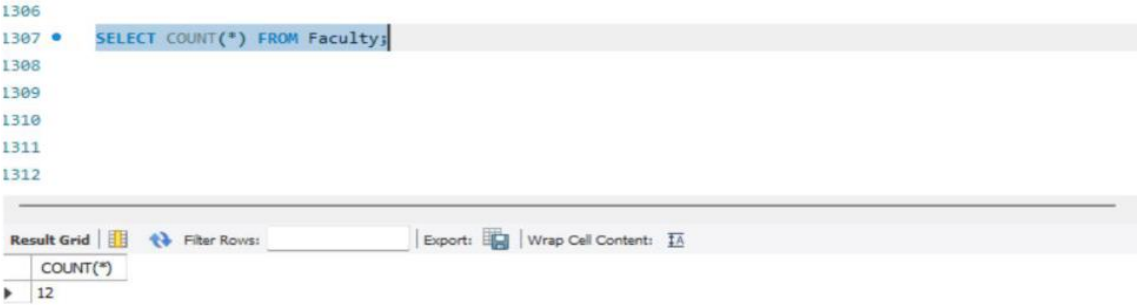
COUNT(*)
8

Faculty:



FacultyID	Lastname	firstname	middleInitial	Email	Contact	SubjectID
12	Naz	Vince Angelo	E	vinceangelo.naz@school.edu	639876859302	5
13	Agsunod	Jerry	B	jerry.agsunod@school.edu	632987621353	1
14	Perete	Paulo	LI	paulo.perete@school.edu	639857282812	2
15	Guillermo	Red	V	red.guillermo@school.edu	632987698764	3
16	Arispe	Maria Charmy	A	maricharmy.arispe@school.edu	632987667890	6
17	Dorosan	Blessica	B	blessica.dorosan@school.edu	632987621353	1
18	Causapin	Suzanne	S	suzanne.causapin@school.edu	632987654321	7
19	Botin	Khristine	A	khristine.botin@school.edu	632987609876	8
20	Aganan	Jorge Sulpicio	S	jorgesulpicio.aganan@school.edu	632987687592	1
21	Ariño	Mary Antoniette	S	maryantoniette.ariño@school.edu	632987687649	3
22	Carinan	Joseph	L	joseph.carinan@school.edu	634563799365	5
23	Platon	Arnold	B	arnold.platon@school.edu	636917941089	2
NULL	NULL	NULL	NULL	NULL	NULL	NULL

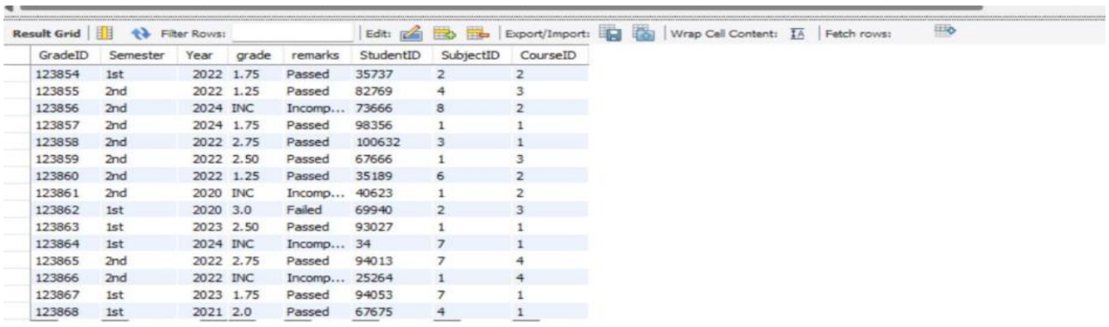
Row Count Per Table:



```
SELECT COUNT(*) FROM Faculty;
```

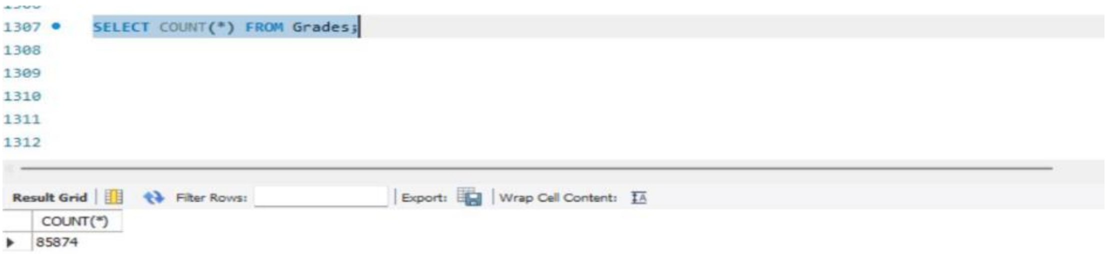
COUNT(*)
12

Grades:



GradeID	Semester	Year	grade	remarks	StudentID	SubjectID	CourseID
123854	1st	2022	1.75	Passed	35737	2	2
123855	2nd	2022	1.25	Passed	82769	4	3
123856	2nd	2024	INC	Incomp...	73666	8	2
123857	2nd	2024	1.75	Passed	98356	1	1
123858	2nd	2022	2.75	Passed	100632	3	1
123859	2nd	2022	2.50	Passed	67666	1	3
123860	2nd	2022	1.25	Passed	35189	6	2
123861	2nd	2020	INC	Incomp...	40623	1	2
123862	1st	2020	3.0	Failed	69940	2	3
123863	1st	2023	2.50	Passed	93027	1	1
123864	1st	2024	INC	Incomp...	34	7	1
123865	2nd	2022	2.75	Passed	94013	7	4
123866	2nd	2022	INC	Incomp...	25264	1	4
123867	1st	2023	1.75	Passed	94053	7	1
123868	1st	2021	2.0	Passed	67675	4	1

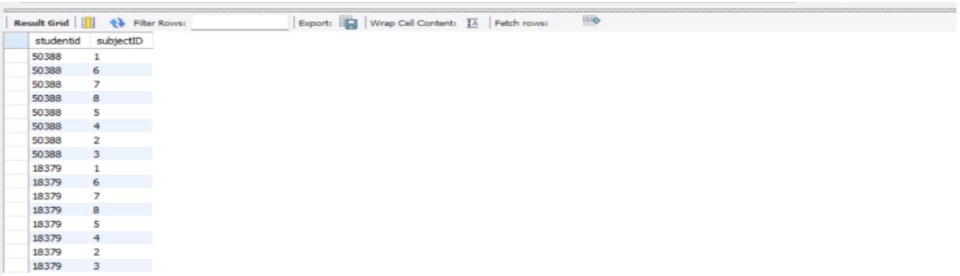
Row Count Per Table:



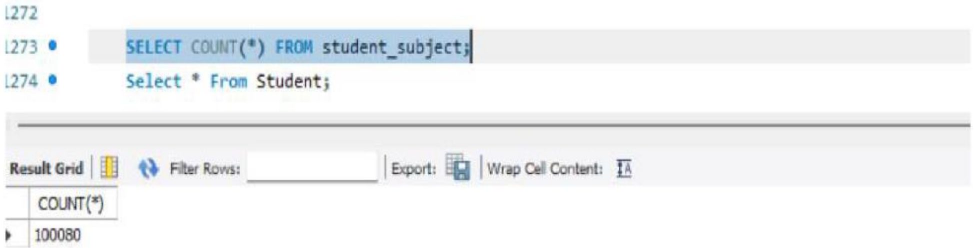
```
SELECT COUNT(*) FROM Grades;
```

COUNT(*)
85874

Student Subject:



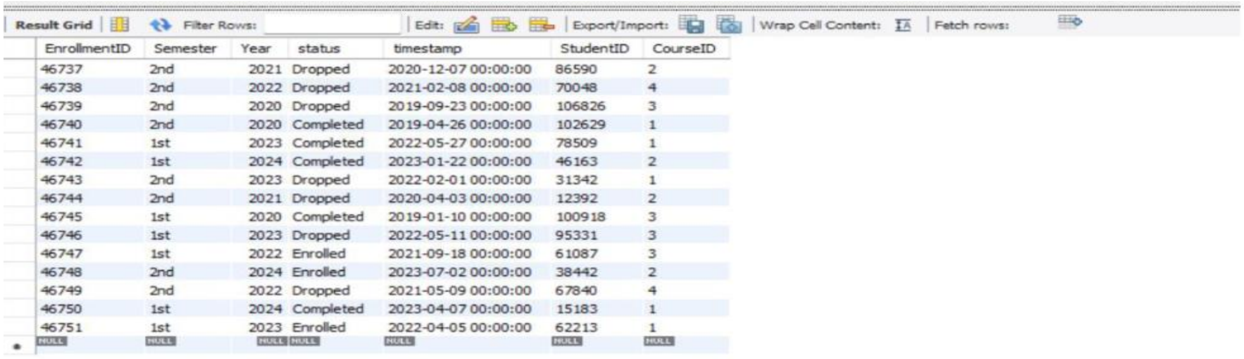
studentid	subjectID
50388	1
50388	6
50388	7
50388	8
50388	5
50388	4
50388	2
50388	3
18379	1
18379	6
18379	7
18379	8
18379	5
18379	4
18379	2
18379	3



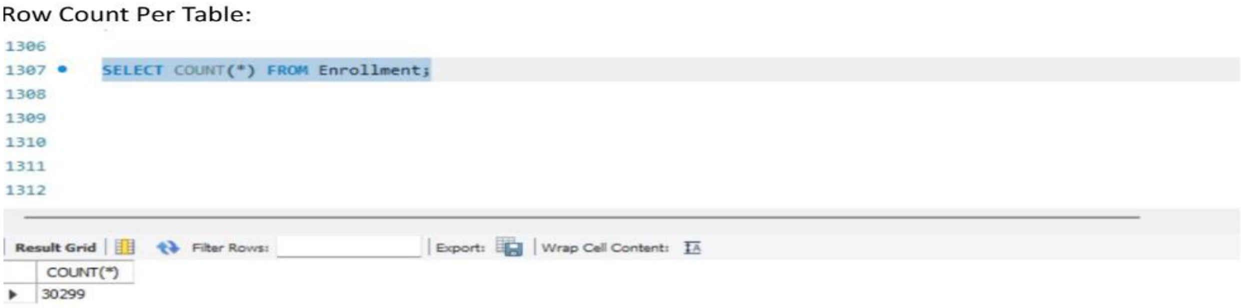
```
SELECT COUNT(*) FROM student_subject;
Select * From Student;
```

COUNT(*)
100080

Enrollment:



EnrollmentID	Semester	Year	status	timestamp	StudentID	CourseID
46737	2nd	2021	Dropped	2020-12-07 00:00:00	86590	2
46738	2nd	2022	Dropped	2021-02-08 00:00:00	70048	4
46739	2nd	2020	Dropped	2019-09-23 00:00:00	106826	3
46740	2nd	2020	Completed	2019-04-26 00:00:00	102629	1
46741	1st	2023	Completed	2022-05-27 00:00:00	78509	1
46742	1st	2024	Completed	2023-01-22 00:00:00	46163	2
46743	2nd	2023	Dropped	2022-02-01 00:00:00	31342	1
46744	2nd	2021	Dropped	2020-04-03 00:00:00	12392	2
46745	1st	2020	Completed	2019-01-10 00:00:00	100918	3
46746	1st	2023	Dropped	2022-05-11 00:00:00	95331	3
46747	1st	2022	Enrolled	2021-09-18 00:00:00	61087	3
46748	2nd	2024	Enrolled	2023-07-02 00:00:00	38442	2
46749	2nd	2022	Dropped	2021-05-09 00:00:00	67840	4
46750	1st	2024	Completed	2023-04-07 00:00:00	15183	1
46751	1st	2023	Enrolled	2022-04-05 00:00:00	62213	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL

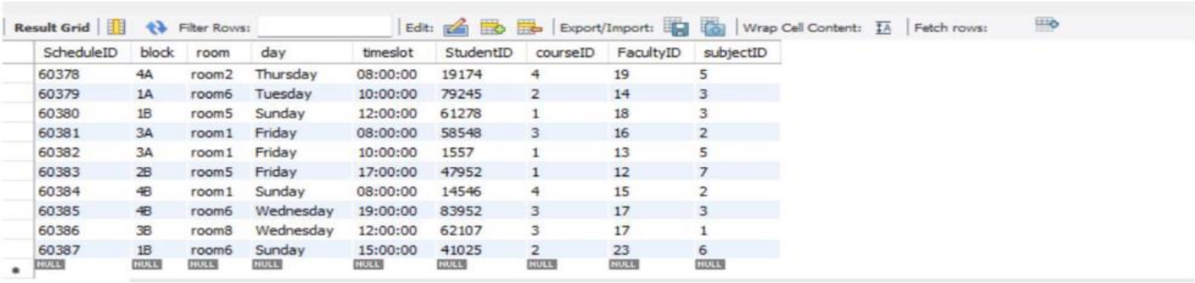


Row Count Per Table:

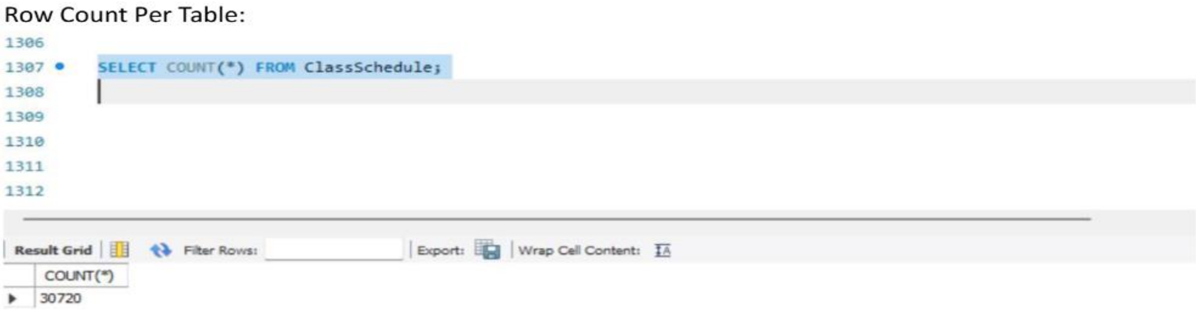
```
SELECT COUNT(*) FROM Enrollment;
```

COUNT(*)
30299

Class Schedule:



ScheduleID	block	room	day	timeslot	StudentID	courseID	FacultyID	subjectID
60378	4A	room2	Thursday	08:00:00	19174	4	19	5
60379	1A	room6	Tuesday	10:00:00	79245	2	14	3
60380	1B	room5	Sunday	12:00:00	61278	1	18	3
60381	3A	room1	Friday	08:00:00	58548	3	16	2
60382	3A	room1	Friday	10:00:00	1557	1	13	5
60383	2B	room5	Friday	17:00:00	47952	1	12	7
60384	4B	room1	Sunday	08:00:00	14546	4	15	2
60385	4B	room6	Wednesday	19:00:00	83952	3	17	3
60386	3B	room8	Wednesday	12:00:00	62107	3	17	1
60387	1B	room6	Sunday	15:00:00	41025	2	23	6
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

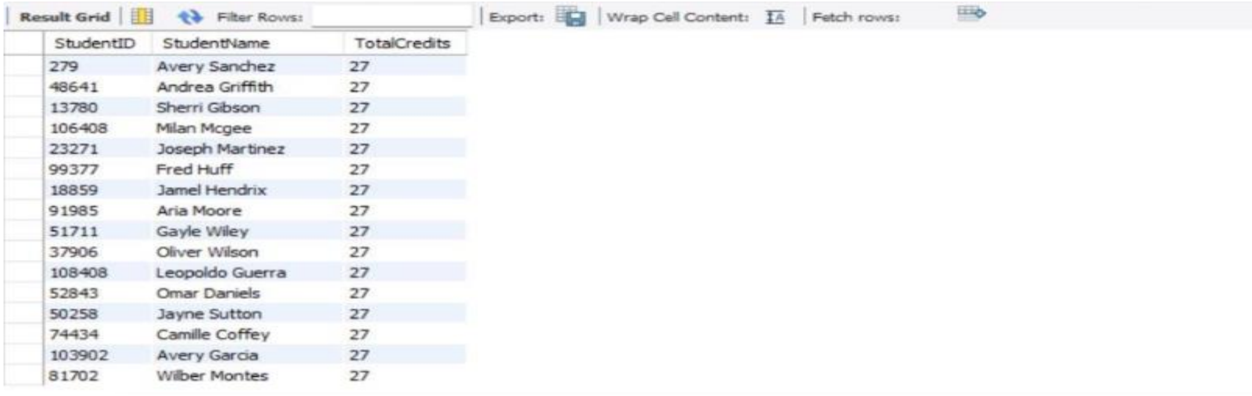


Row Count Per Table:

```
SELECT COUNT(*) FROM ClassSchedule;
```

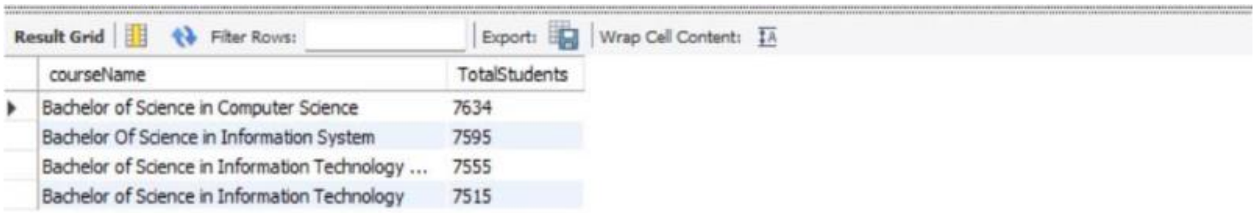
COUNT(*)
30720

Total Credits Earned by Each Student (SUM)



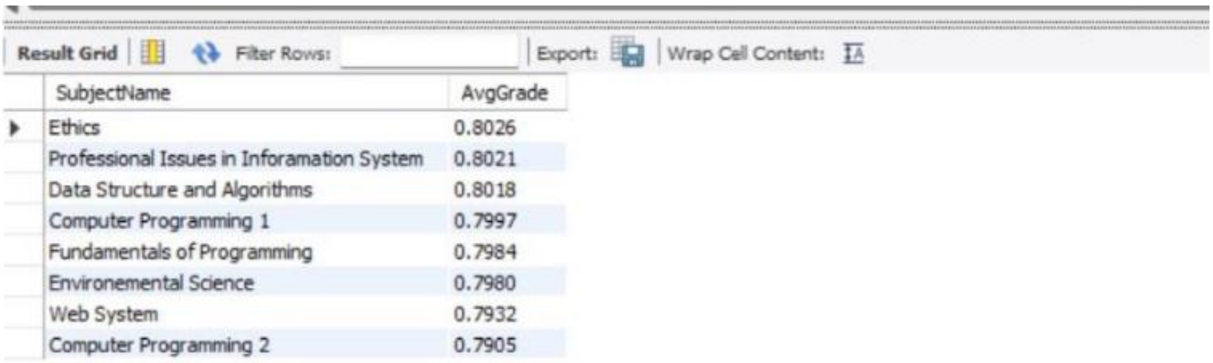
StudentID	StudentName	TotalCredits
279	Avery Sanchez	27
48641	Andrea Griffith	27
13780	Sherri Gibson	27
106408	Milan Mcgee	27
23271	Joseph Martinez	27
99377	Fred Huff	27
18859	Jamel Hendrix	27
91985	Aria Moore	27
51711	Gayle Wiley	27
37906	Oliver Wilson	27
108408	Leopoldo Guerra	27
52843	Omar Daniels	27
50258	Jayne Sutton	27
74434	Camille Coffey	27
103902	Avery Garcia	27
81702	Wilber Montes	27

Number of Students Enrolled Per Course (COUNT)



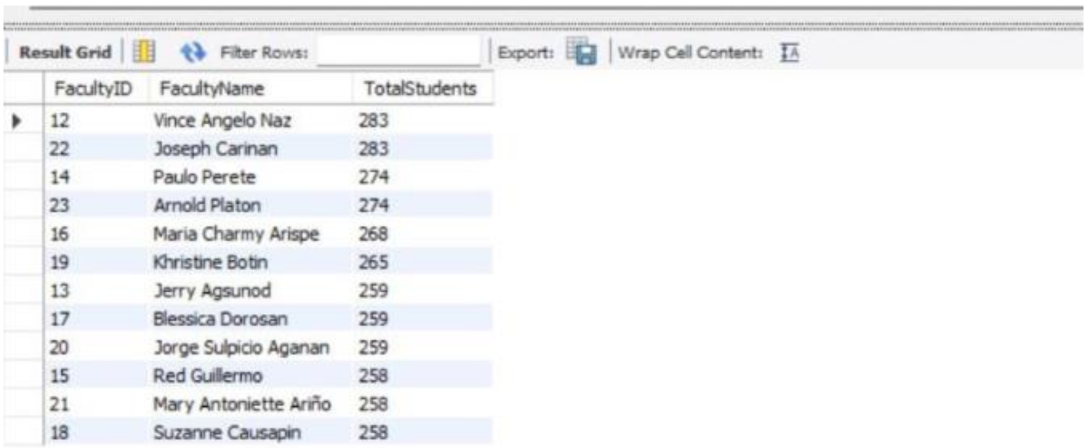
courseName	TotalStudents
Bachelor of Science in Computer Science	7634
Bachelor Of Science in Information System	7595
Bachelor of Science in Information Technology ...	7555
Bachelor of Science in Information Technology	7515

Average Grade Per Subject (AVG)



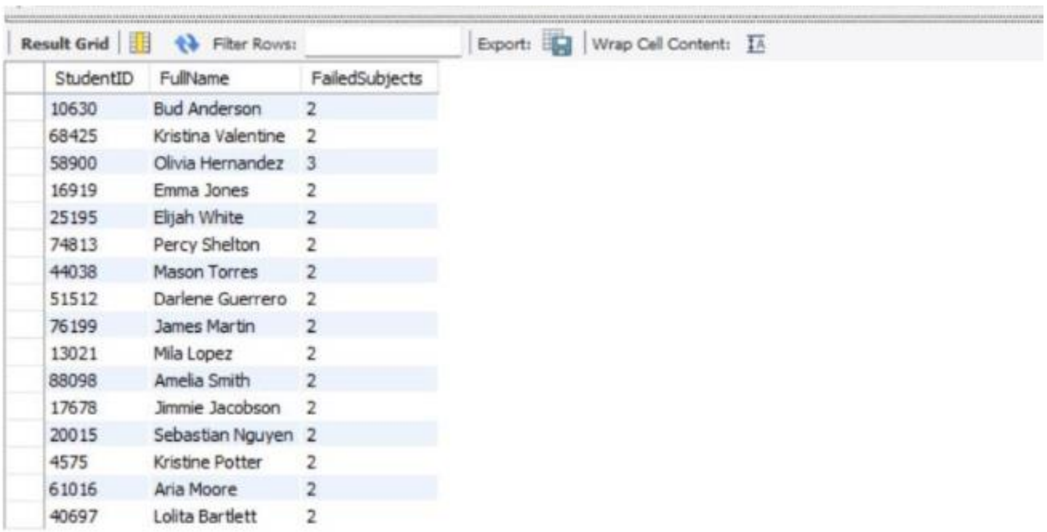
SubjectName	AvgGrade
Ethics	0.8026
Professional Issues in Inforamation System	0.8021
Data Structure and Algorithms	0.8018
Computer Programming 1	0.7997
Fundamentals of Programming	0.7984
Environemental Science	0.7980
Web System	0.7932
Computer Programming 2	0.7905

Faculty Members with Most Enrolled Students (JOIN + COUNT)



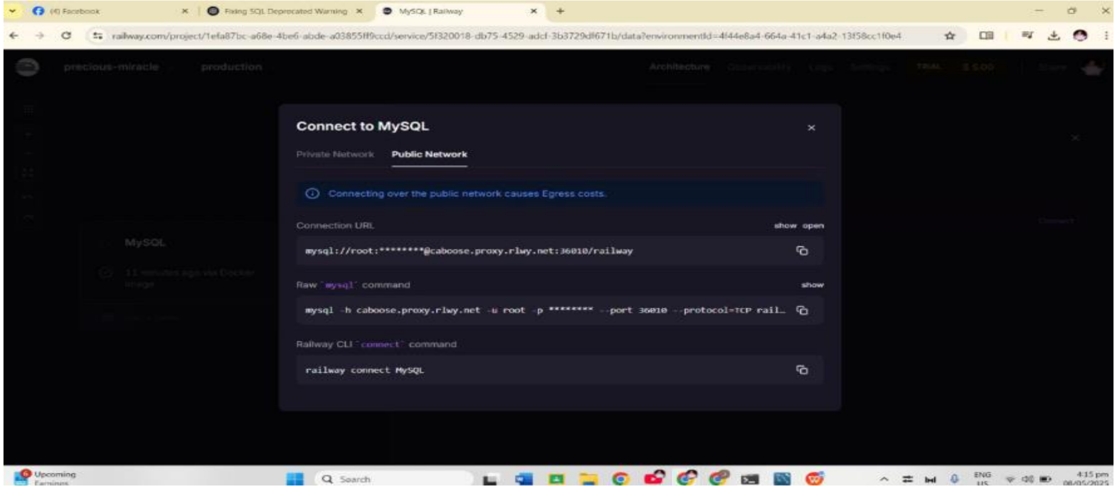
FacultyID	FacultyName	TotalStudents
12	Vince Angelo Naz	283
22	Joseph Carinan	283
14	Paulo Perete	274
23	Arnold Platon	274
16	Maria Charmy Arispe	268
19	Khristine Botin	265
13	Jerry Agsunod	259
17	Blessica Dorosan	259
20	Jorge Sulpicio Aganan	259
15	Red Guillermo	258
21	Mary Antoniette Ariño	258
18	Suzanne Causapin	258

Students Who Failed More Than One Subject



StudentID	FullName	FailedSubjects
10630	Bud Anderson	2
68425	Kristina Valentine	2
58900	Olivia Hernandez	3
16919	Emma Jones	2
25195	Elijah White	2
74813	Percy Shelton	2
44038	Mason Torres	2
51512	Darlene Guerrero	2
76199	James Martin	2
13021	Mila Lopez	2
88098	Amelia Smith	2
17678	Jimmie Jacobson	2
20015	Sebastian Nguyen	2
4575	Kristine Potter	2
61016	Aria Moore	2
40697	Lolita Bartlett	2

Documentation of cloud deployment:



Connect to MySQL

Private Network | Public Network

Connecting over the public network causes Egress costs.

Connection URL

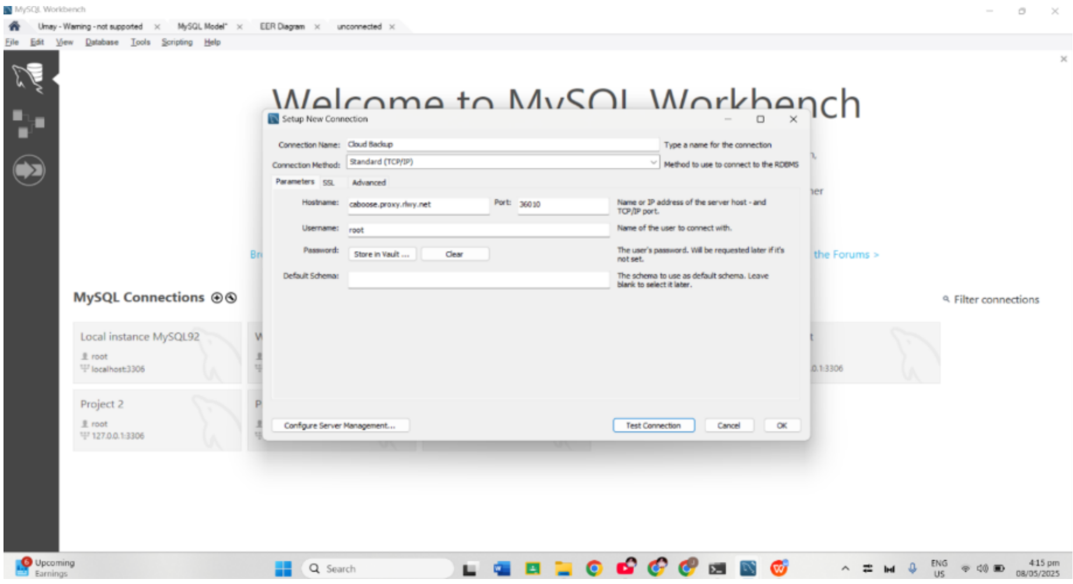
```
mysql://root:********@caboose.proxy.rlwy.net:36010/railway
```

Raw `mysql` command

```
mysql -h caboose.proxy.rlwy.net -u root -p ******** --port 36010 --protocol=TCP rail…
```

Railway CLI `connect` command

```
railway connect MySQL
```



Setup New Connection

Connection Name: Cloud Backup — Type a name for the connection

Connection Method: Standard (TCP/IP) — Method to use to connect to the RDBMS

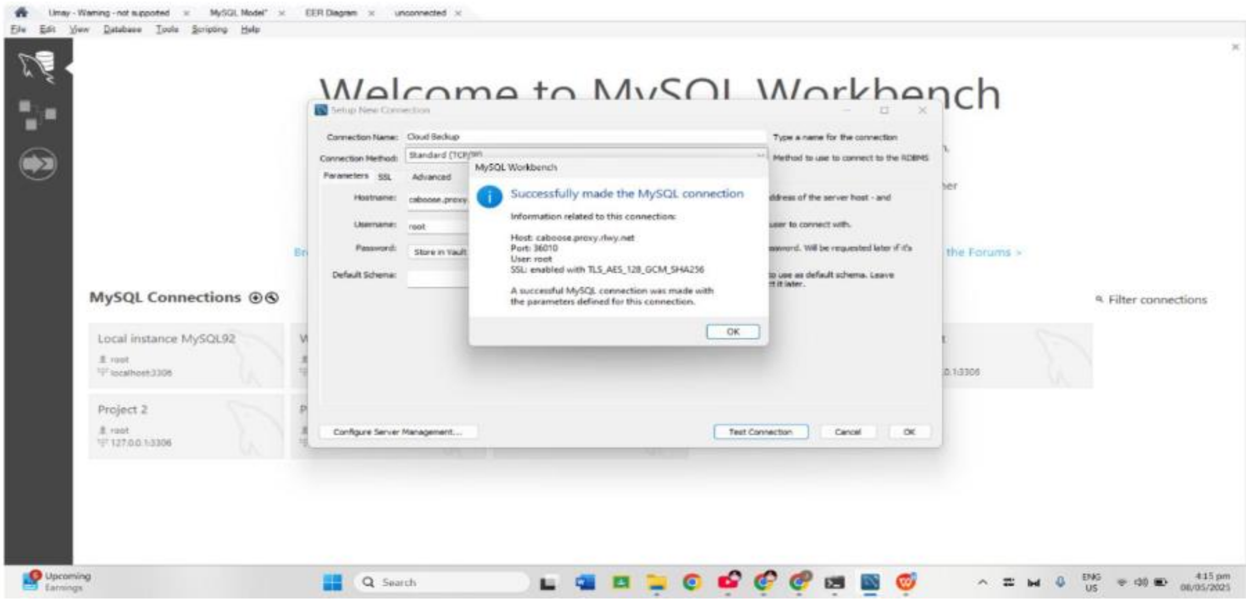
Parameters | SSL | Advanced

Hostname: caboose.proxy.rlwy.net Port: 36010 — Name or IP address of the server host - and TCP/IP port.

Username: root — Name of the user to connect with.

Password: Store in Vault ... Clear — The user's password. Will be requested later if it's not set.

Default Schema: — The schema to use as default schema. Leave blank to select it later.



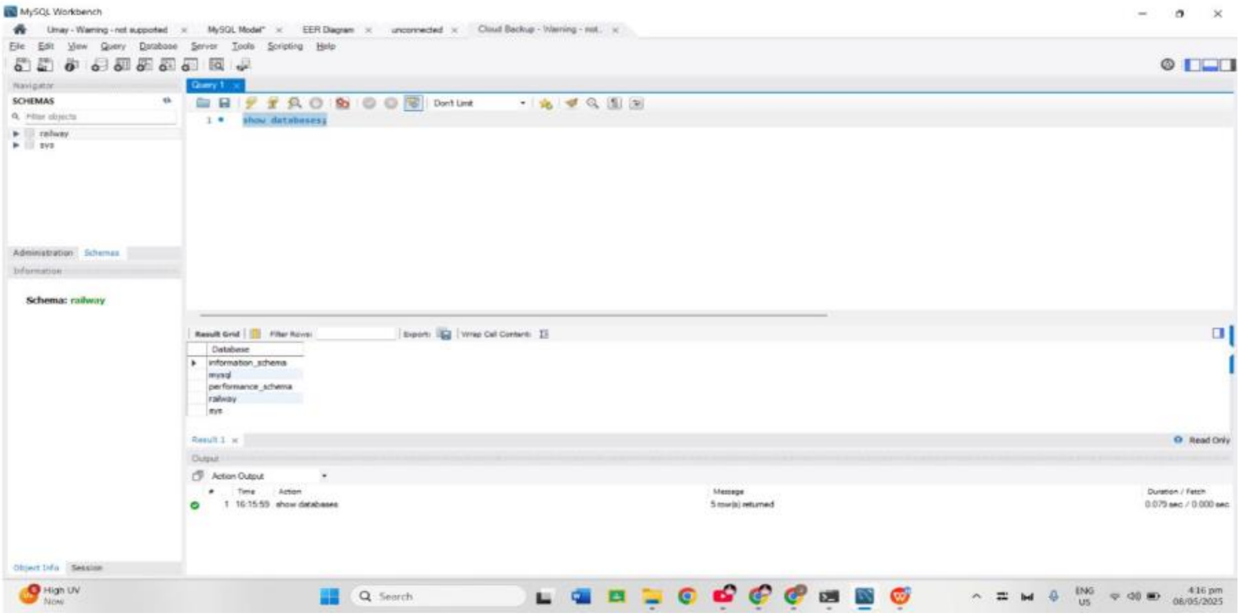
MySQL Workbench

Successfully made the MySQL connection

Information related to this connection:

Host: caboose.proxy.rlwy.net
Port: 36010
User: root
SSL: enabled with TLS_AES_128_GCM_SHA256

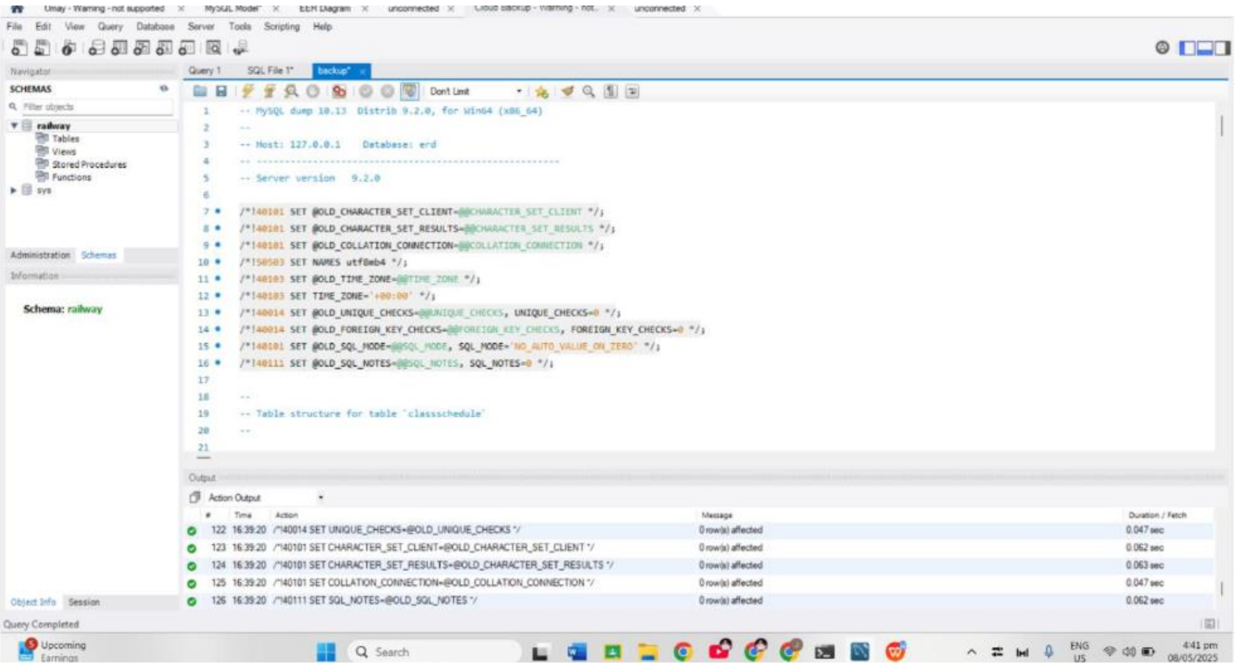
A successful MySQL connection was made with the parameters defined for this connection.



```
show databases;
```

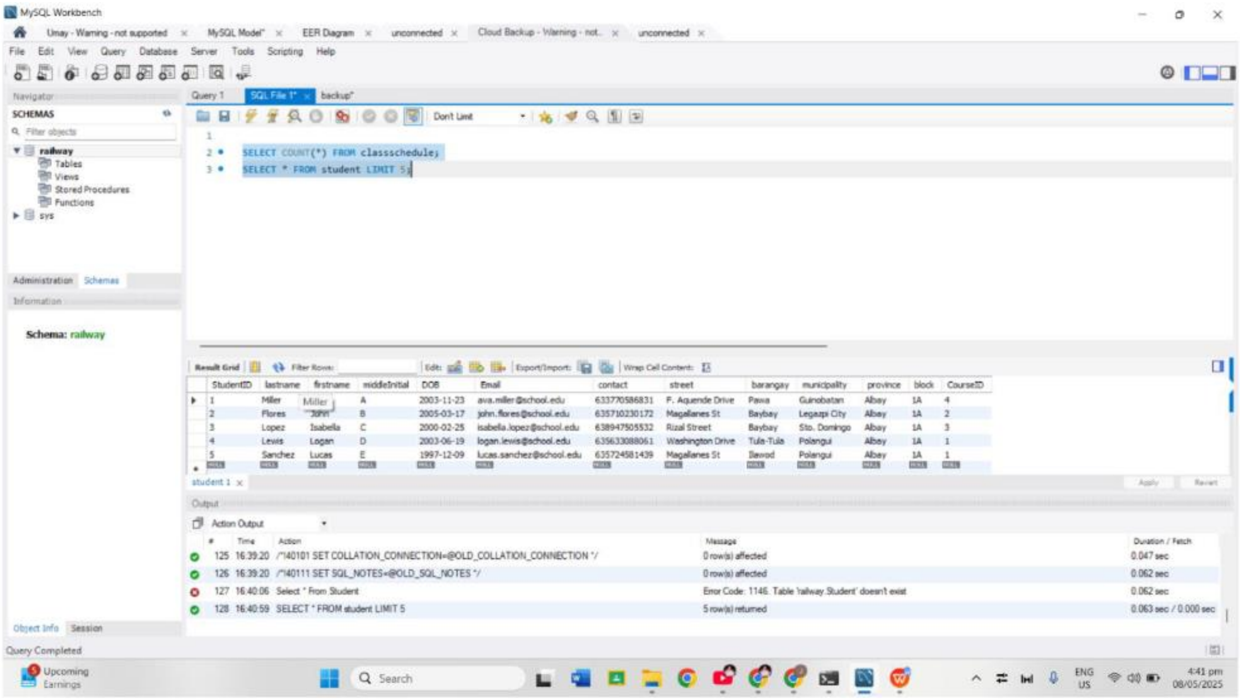
Database
information_schema
mysql
performance_schema
railway
sys

```
#   Time      Action           Message            Duration / Fetch
1   16:15:59  show databases   5 row(s) returned  0.079 sec / 0.000 sec
```



```
-- MySQL dump 10.13  Distrib 9.2.0, for Win64 (x86_64)
--
-- Host: 127.0.0.1    Database: erd
-- ------------------------------------------------------
-- Server version   9.2.0

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!50503 SET NAMES utf8mb4 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `classschedule`
--
```

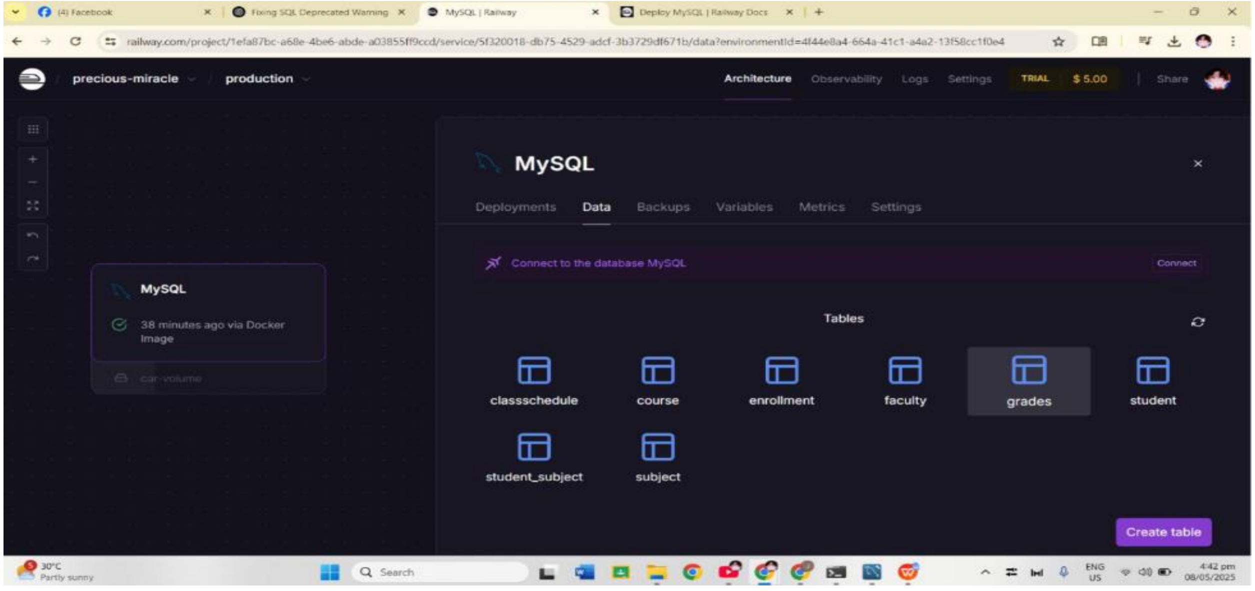
```
#    Time      Action                                                           Message            Duration / Fetch
122  16:39:20  /*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */                  0 row(s) affected  0.047 sec
123  16:39:20  /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */    0 row(s) affected  0.062 sec
124  16:39:20  /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */  0 row(s) affected  0.063 sec
125  16:39:20  /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */    0 row(s) affected  0.047 sec
126  16:39:20  /*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */                          0 row(s) affected  0.062 sec
```



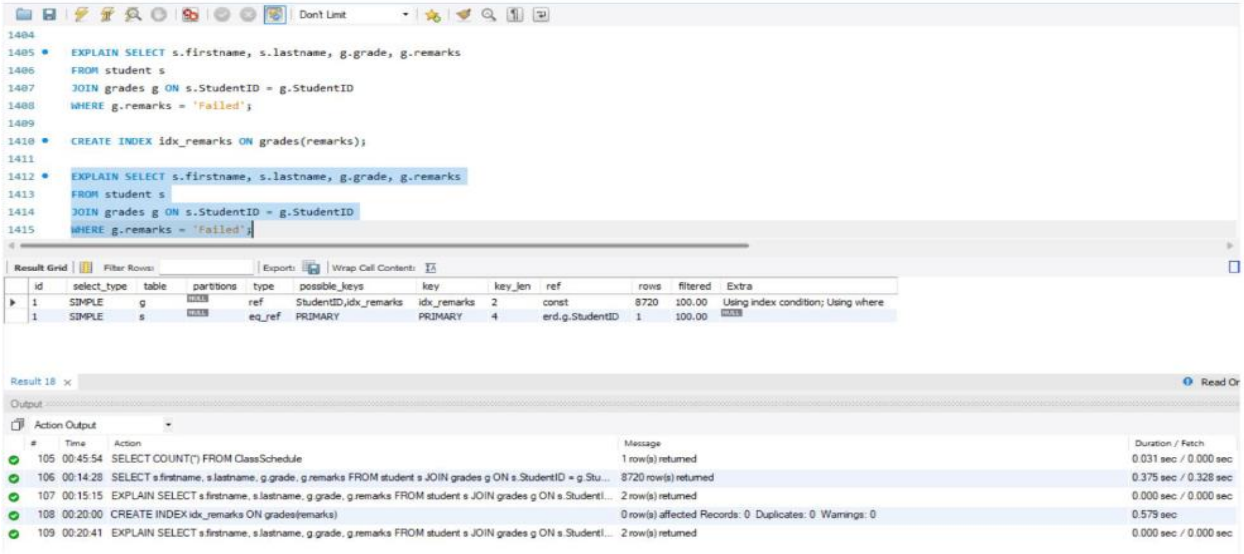
```
SELECT COUNT(*) FROM classschedule;
SELECT * FROM student LIMIT 5;
```

StudentID	lastname	firstname	middleInitial	DOB	Email	contact	street	barangay	municipality	province	block	CourseID
1	Miller	[illegible]	A	2003-11-23	ava.miller@school.edu	633770586831	F. Aquende Drive	Pawa	Guinobatan	Albay	1A	4
2	Flores	John	B	2005-03-17	john.flores@school.edu	635710230172	Magallanes St	Baybay	Legazpi City	Albay	1A	2
3	Lopez	Isabella	C	2000-02-25	isabella.lopez@school.edu	638947505532	Rizal Street	Baybay	Sto. Domingo	Albay	1A	3
4	Lewis	Logan	D	2003-06-19	logan.lewis@school.edu	635633088061	Washington Drive	Tula-Tula	Polangui	Albay	1A	1
5	Sanchez	Lucas	E	1997-12-09	lucas.sanchez@school.edu	635724581439	Magallanes St	Ilawod	Polangui	Albay	1A	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
#    Time      Action                                                         Message                                          Duration / Fetch
125  16:39:20  /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */  0 row(s) affected                                0.047 sec
126  16:39:20  /*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */                        0 row(s) affected                                0.062 sec
127  16:40:06  Select * From Student                                           Error Code: 1146. Table 'railway.Student' doesn't exist  0.062 sec
128  16:40:59  SELECT * FROM student LIMIT 5                                   5 row(s) returned                                0.063 sec / 0.000 sec
```



After Indexing:

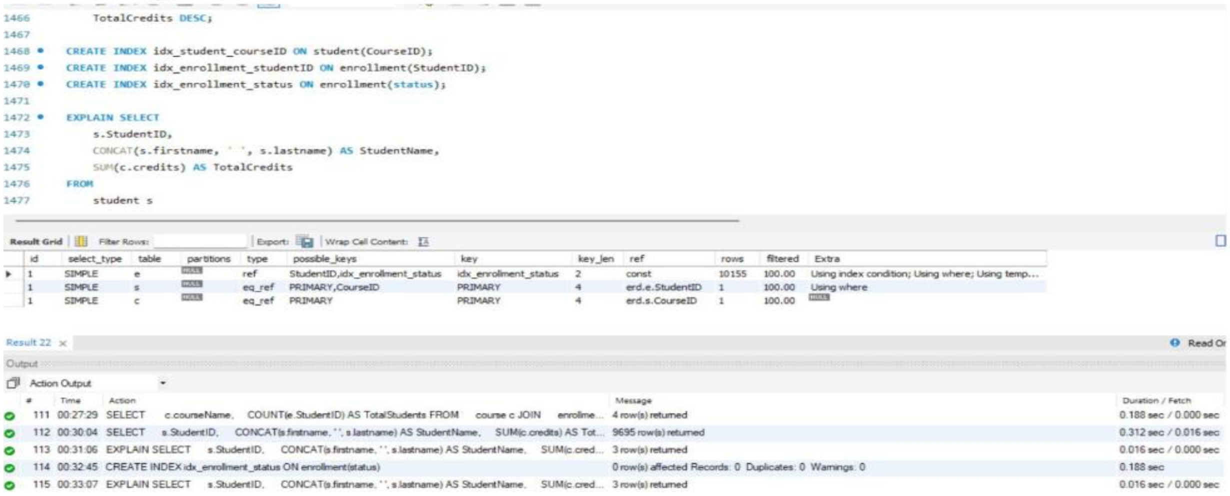


```
EXPLAIN SELECT s.firstname, s.lastname, g.grade, g.remarks
FROM student s
JOIN grades g ON s.StudentID = g.StudentID
WHERE g.remarks = 'Failed';

CREATE INDEX idx_remarks ON grades(remarks);

EXPLAIN SELECT s.firstname, s.lastname, g.grade, g.remarks
FROM student s
JOIN grades g ON s.StudentID = g.StudentID
WHERE g.remarks = 'Failed';
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	g	NULL	ref	StudentID,idx_remarks	idx_remarks	2	const	8720	100.00	Using index condition; Using where
1	SIMPLE	s	NULL	eq_ref	PRIMARY	PRIMARY	4	erd.g.StudentID	1	100.00	NULL

```
#    Time      Action                                                                            Message                                                     Duration / Fetch
105  00:45:54  SELECT COUNT(*) FROM ClassSchedule                                                1 row(s) returned                                           0.031 sec / 0.000 sec
106  00:14:28  SELECT s.firstname, s.lastname, g.grade, g.remarks FROM student s JOIN grades g ON s.StudentID = g.Stu...  8720 row(s) returned       0.375 sec / 0.328 sec
107  00:15:15  EXPLAIN SELECT s.firstname, s.lastname, g.grade, g.remarks FROM student s JOIN grades g ON s.StudentI...  2 row(s) returned          0.000 sec / 0.000 sec
108  00:20:00  CREATE INDEX idx_remarks ON grades(remarks)                                       0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0   0.579 sec
109  00:20:41  EXPLAIN SELECT s.firstname, s.lastname, g.grade, g.remarks FROM student s JOIN grades g ON s.StudentI...  2 row(s) returned          0.000 sec / 0.000 sec
```

After Indexing:

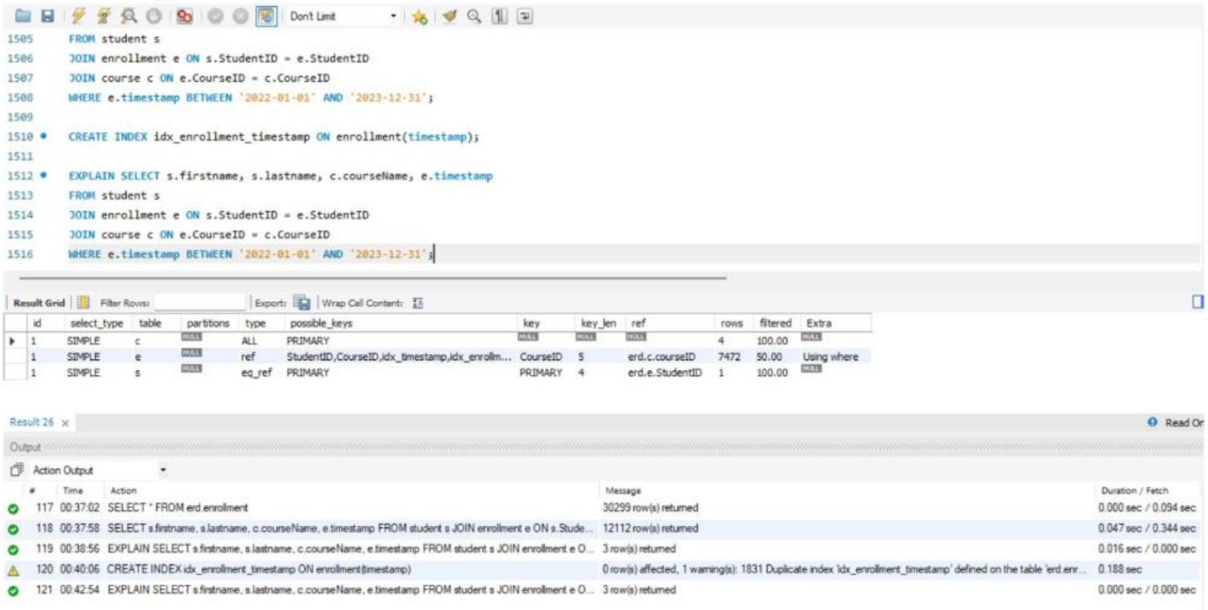


```
    TotalCredits DESC;

CREATE INDEX idx_student_courseID ON student(CourseID);
CREATE INDEX idx_enrollment_studentID ON enrollment(StudentID);
CREATE INDEX idx_enrollment_status ON enrollment(status);

EXPLAIN SELECT
    s.StudentID,
    CONCAT(s.firstname, ' ', s.lastname) AS StudentName,
    SUM(c.credits) AS TotalCredits
FROM
    student s
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	e	NULL	ref	StudentID,idx_enrollment_status	idx_enrollment_status	2	const	10155	100.00	Using index condition; Using where; Using temp...
1	SIMPLE	s	NULL	eq_ref	PRIMARY,CourseID	PRIMARY	4	erd.e.StudentID	1	100.00	Using where
1	SIMPLE	c	NULL	eq_ref	PRIMARY	PRIMARY	4	erd.s.CourseID	1	100.00	NULL

```
#    Time      Action                                                                                  Message                                                     Duration / Fetch
111  00:27:29  SELECT    c.courseName,   COUNT(e.StudentID) AS TotalStudents FROM    course c JOIN    enrollme...  4 row(s) returned        0.188 sec / 0.000 sec
112  00:30:04  SELECT    s.StudentID,   CONCAT(s.firstname, ' ', s.lastname) AS StudentName,   SUM(c.credits) AS Tot...  9695 row(s) returned     0.312 sec / 0.016 sec
113  00:31:06  EXPLAIN SELECT    s.StudentID,   CONCAT(s.firstname, ' ', s.lastname) AS StudentName,   SUM(c.cred...  3 row(s) returned      0.016 sec / 0.000 sec
114  00:32:45  CREATE INDEX idx_enrollment_status ON enrollment(status)                                 0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0   0.188 sec
115  00:33:07  EXPLAIN SELECT    s.StudentID,   CONCAT(s.firstname, ' ', s.lastname) AS StudentName,   SUM(c.cred...  3 row(s) returned      0.016 sec / 0.000 sec
```

After Indexing:



```sql
FROM student s
JOIN enrollment e ON s.StudentID = e.StudentID
JOIN course c ON e.CourseID = c.CourseID
WHERE e.timestamp BETWEEN '2022-01-01' AND '2023-12-31';

CREATE INDEX idx_enrollment_timestamp ON enrollment(timestamp);

EXPLAIN SELECT s.firstname, s.lastname, c.courseName, e.timestamp
FROM student s
JOIN enrollment e ON s.StudentID = e.StudentID
JOIN course c ON e.CourseID = c.CourseID
WHERE e.timestamp BETWEEN '2022-01-01' AND '2023-12-31';
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	c	NULL	ALL	PRIMARY	NULL	NULL	NULL	4	100.00	NULL
1	SIMPLE	e	NULL	ref	StudentID,CourseID,idx_timestamp,idx_enrollm...	CourseID	5	erd.c.courseID	7472	50.00	Using where
1	SIMPLE	s	NULL	eq_ref	PRIMARY	PRIMARY	4	erd.e.StudentID	1	100.00	NULL

```
#    Time      Action                                                                                          Message                                                                                              Duration / Fetch
117  00:37:02  SELECT * FROM erd.enrollment                                                                    30299 row(s) returned                                                                                0.000 sec / 0.094 sec
118  00:37:58  SELECT s.firstname, s.lastname, c.courseName, e.timestamp FROM student s JOIN enrollment e ON s.Stude...  12112 row(s) returned                                                                         0.047 sec / 0.344 sec
119  00:38:56  EXPLAIN SELECT s.firstname, s.lastname, c.courseName, e.timestamp FROM student s JOIN enrollment e O...  3 row(s) returned                                                                              0.016 sec / 0.000 sec
120  00:40:06  CREATE INDEX idx_enrollment_timestamp ON enrollment(timestamp)                                  0 row(s) affected, 1 warning(s): 1831 Duplicate index 'idx_enrollment_timestamp' defined on the table 'erd.enr...  0.188 sec
121  00:42:54  EXPLAIN SELECT s.firstname, s.lastname, c.courseName, e.timestamp FROM student s JOIN enrollment e O...  3 row(s) returned                                                                              0.000 sec / 0.000 sec
```

Reflection and limitations

Reflection As a group of college students developing the Registrar Student Management System, we gained valuable hands-on experience in designing and implementing a real-world database solution. Through this project, we deepened our understanding of database structures, SQL syntax, data relationships, and how systems like this support the daily operations of a school registrar. We learned how to create and populate tables that reflect real entities in a school setting students, courses, subjects, grades, and faculty. Writing SQL queries for data analytics, such as course popularity and faculty workload, challenged us to think beyond simple data storage and consider how our system could provide insights to help school administrators make informed decisions. We also explored performance tuning through indexing and query analysis using EXPLAIN, which gave us a better understanding of database optimization. Most importantly, this project showed us the value of planning and structuring our database carefully. It taught us to be mindful of data integrity and the importance of writing efficient, readable SQL code. Overall, this experience strengthened both our technical skills and our ability to work collaboratively on a system that mimics a real registrar environment. Limitations Despite the success of our project, we also encountered several limitations that highlighted areas for improvement: Data Accuracy and Integrity Since we used ORDER BY RAND() and LIMIT to randomly assign foreign key values, there's a chance that the data isn't fully realistic or representative of actual student-course relationships. In a production environment, this could lead to data integrity issues. Simplified Structure Some of our table designs are simplified. For example, the faculty table currently assigns only one subject per faculty member, which may not reflect the real-world scenario where teachers handle multiple subjects or courses. Lack of Constraints and Validation While our database uses primary keys, we haven't enforced foreign key constraints or input validation rules. This means incorrect or inconsistent data could be inserted unless handled by application-level checks. Limited User Roles and Security Our system focuses solely on the data layer. We haven't implemented user roles (e.g., admin, student, registrar staff) or any security measures, which are crucial in real systems to protect sensitive information. Scalability and Real-Time Use The system hasn't been tested under large-scale or real-time conditions. We haven't addressed concerns like concurrent access, transactional safety, or long-term scalability. No Front-End Integration Since our work was mostly on the backend with SQL, there's no user-friendly interface yet. This makes the system harder to use for non-technical users like actual registrar staff or students.