Class Quiz Report

Prepared by Group X (3 Students)

Course: [Your Course Name]

Date: [Insert Date]

# Table of Contents

1. 1. Introduction
2. 2. Table Creation with Constraints
3. 3. Join Queries (INNER, LEFT, RIGHT, FULL)
4. 4. Index Creation
5. 5. View Creation
6. 6. Results and Discussion
7. 7. Conclusion

# 1. Introduction

In this quiz, our group designed a small relational database to demonstrate key SQL concepts: creating tables with constraints, performing various joins, creating an index, creating a view, and generating a short report of our work.

# 2. Table Creation with Constraints

We created three tables:  
- Students – stores student information  
- Courses – stores course information  
- Enrollments – maps students to courses

SQL Used:

CREATE TABLE Students (  
 StudentID INT PRIMARY KEY,  
 StudentName VARCHAR(50) NOT NULL,  
 Email VARCHAR(50) UNIQUE  
);  
  
CREATE TABLE Courses (  
 CourseID INT PRIMARY KEY,  
 CourseName VARCHAR(50) NOT NULL  
);  
  
CREATE TABLE Enrollments (  
 EnrollmentID INT PRIMARY KEY,  
 StudentID INT,  
 CourseID INT,  
 FOREIGN KEY (StudentID) REFERENCES Students(StudentID),  
 FOREIGN KEY (CourseID) REFERENCES Courses(CourseID)  
);

(Insert screenshot of table creation here)

# 3. Join Queries

We demonstrated:  
- INNER JOIN – Students enrolled in courses  
- LEFT JOIN – All students with their courses (if any)  
- RIGHT JOIN – All courses and which students are enrolled  
- FULL OUTER JOIN – All students and all courses (if supported by DB)

Example:

SELECT s.StudentName, c.CourseName  
FROM Students s  
INNER JOIN Enrollments e ON s.StudentID = e.StudentID  
INNER JOIN Courses c ON e.CourseID = c.CourseID;

(Insert screenshots or output tables here)

# 4. Index Creation

We created an index to improve query performance on frequently searched columns:

CREATE INDEX idx\_studentname ON Students(StudentName);

(Insert screenshot of index creation here)

# 5. View Creation

We simplified data access by creating a view:

CREATE OR REPLACE VIEW StudentCourseView AS  
SELECT s.StudentName, c.CourseName  
FROM Students s  
INNER JOIN Enrollments e ON s.StudentID = e.StudentID  
INNER JOIN Courses c ON e.CourseID = c.CourseID;

Now we can simply query:

SELECT \* FROM StudentCourseView;

(Insert screenshot of view creation and SELECT results here)

# 6. Results and Discussion

We successfully:  
- Created tables with constraints.  
- Executed different join queries and displayed results.  
- Created an index to improve query performance.  
- Created a view to simplify access to combined data.  
  
The view returns student names with their course names directly, which is easier than writing multiple JOINs each time.

# 7. Conclusion

This exercise demonstrated how to:  
- Model data with relationships and constraints.  
- Retrieve data using different types of joins.  
- Optimize performance with indexes.  
- Simplify queries with views.