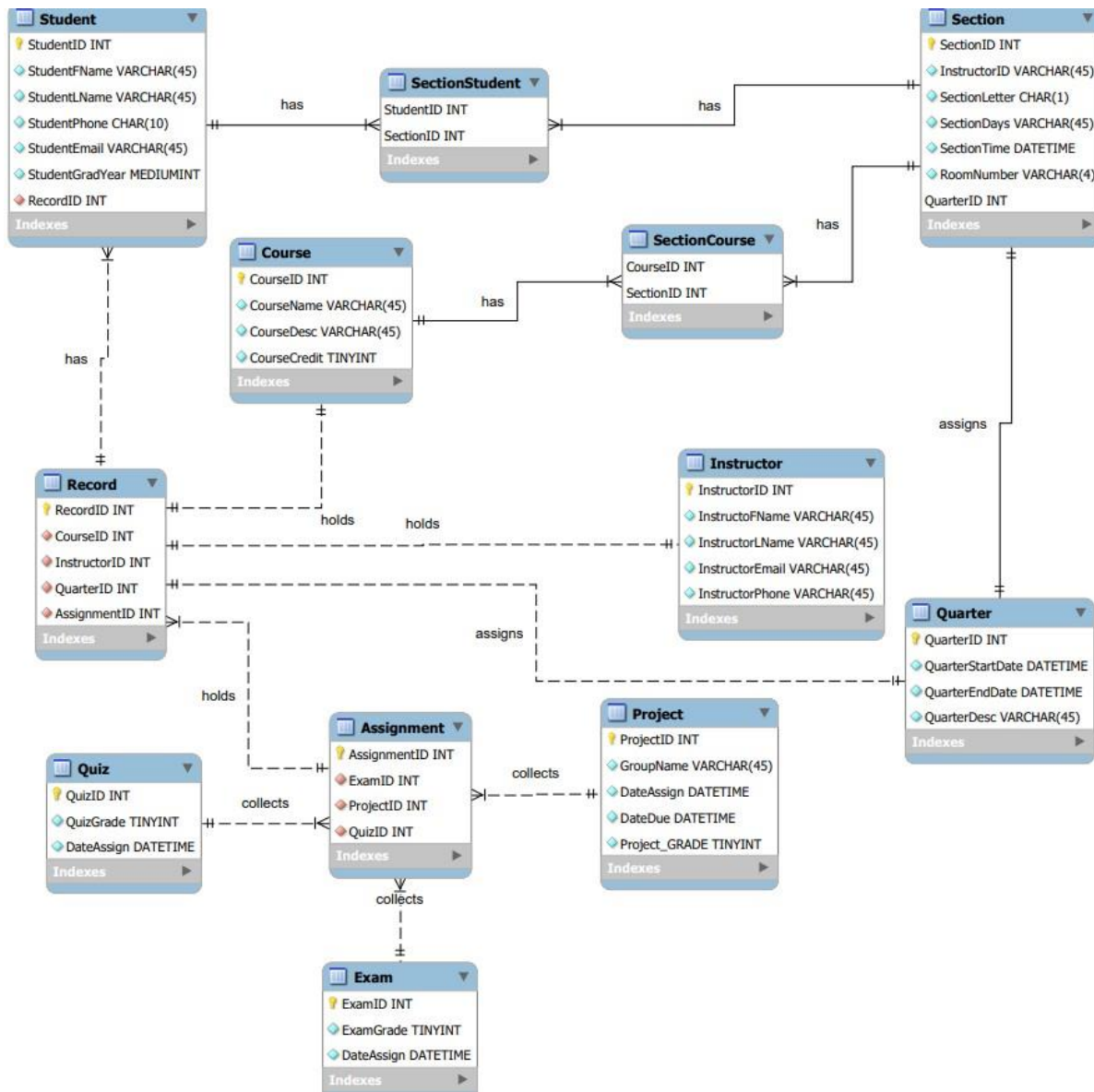


Updated ER Diagram



-- 12 SQL Queries -

- #1 Show the average exam grade.

```
SELECT AVG(ExamGrade)
FROM Exam;
```

AVG(ExamGrade)
83.2000

-- #2 List all of the instructors that work in the "A" building, sorted by room number.

```
SELECT DISTINCT InstructorLName, InstructorFName
FROM Section JOIN Instructor USING(InstructorID)
WHERE RoomNumber LIKE 'A%'
ORDER BY RoomNumber;
```

	InstructorLName	InstructorFName
▶	Galgey	Leon
	Stutte	Gunther
	Durnall	Lin

-- #3 List all class times in the room A202.

```
SELECT RoomNumber, SectionDays, SectionTime
FROM Section
WHERE RoomNumber = "A202" ORDER BY SectionTime DESC;
```

	RoomNumber	SectionDays	SectionTime
▶	A202	TTH	9:15 AM
	A202	TTH	8:00 AM

#4 Find the contact information for instructors with the last name "Lidgate".

```
SELECT InstructorLName, InstructorFName, InstructorEmail, InstructorPhone
FROM Instructor
WHERE InstructorLNAME = "Durnall";
```

	InstructorLName	InstructorFName	InstructorEmail	InstructorPhone
▶	Durnall	Lin	ldurnall0@techcrunch.com	757-707-0088

-- #5 List the courses offered in the Winter quarter.

```
SELECT QuarterDesc, CourseName
FROM Quarter JOIN Section USING(QuarterID) JOIN Course USING(CourseID)
WHERE QuarterDesc = "Winter";
```

--

	QuarterDesc	CourseName
►	Winter	EKQZ9038
	Winter	BYLZ7557
	Winter	MQTP8095
	Winter	JTIP1556
	Winter	HMBC1693

-- #6 Return the number of students that have graduated before 2020.

```
SELECT COUNT(StudentID)
FROM Student
WHERE StudentGradYear < 2020;
```

	COUNT(StudentID)
►	7

-- #7 Return the name of students that were in the group named "Gorilla gorilla".

```
SELECT CONCAT(StudentFNAME, ' ', StudentLNAME) StudentFULLNAME
FROM Student JOIN Record USING(RecordID) JOIN Assignment USING (AssignmentID)
JOIN Project USING (ProjectID)
WHERE GroupName = 'Gorilla gorilla' ORDER BY StudentFULLNAME;
```

	StudentFULLNAME
►	Pat Skottle

--

#8 List the exam history of the student with the ID '41908376'.

```
SELECT RecordID, AssignmentID, ExamID, ExamGrade
FROM Student JOIN Record USING(RecordID) JOIN Assignment USING(AssignmentID)
JOIN Exam USING(ExamID)
WHERE StudentID = 41908376;
```

	RecordID	AssignmentID	ExamID	ExamGrade
▶	84562133	87482923	11844339	91

-- #9 Return the class schedule for the student(s) named "Tilly Lepope".

```
SELECT StudentID, SectionID, CourseName, CourseCredit, SectionDays, SectionTime,
RoomNumber
FROM Student JOIN SectionStudent USING(StudentID) JOIN Section USING(SectionID) JOIN
Course USING(CourseID)
WHERE StudentFName = "Tilly" AND StudentLName = "Lepope";
```

	StudentID	SectionID	CourseName	CourseCredit	SectionDays	SectionTime	RoomNumber
▶	41908376	49051067	DCVN6135	3	TTH	10:55 AM	C605

-- #10 Return the names of all students in the section with ID '17009066', ordered by last name.

```
SELECT SectionID, StudentID, StudentLName, StudentFName
FROM Section JOIN SectionStudent USING(SectionID) JOIN Student USING(StudentID)
WHERE SECTIONID = 17009066
ORDER BY StudentLName;
```

#11 Update first name with ID = 61749685.

```
UPDATE Student
SET StudentFName = 'Nathan', StudentGradYear = 2011
WHERE StudentID = 61749685;
```

[illegible]

--

#12 Return the names of the graduated students with AVG(ExamGrade) >= 90, order by full name.

```
SELECT CONCAT(StudentFNAME, ' ', StudentLNAME) StudentFullName, RecordID,
AssignmentID, ExamID, ROUND(AVG(ExamGrade),2) AS AVG_ExamGrade
FROM Student JOIN Record USING(RecordID) JOIN Assignment USING(AssignmentID)
JOIN Exam USING (ExamID) JOIN Quarter USING (QuarterID)
WHERE StudentGradYear <= 2021
GROUP BY CONCAT(StudentFNAME, ' ', StudentLNAME), RecordID, AssignmentID,
ExamID
HAVING AVG_ExamGrade >= 90
ORDER BY StudentFullName;
```

	StudentFullName	RecordID	AssignmentID	ExamID	AVG_ExamGrade
▶	Avis Denziloe	9867772	74276428	69856066	90.00
	Mirelle Blodgett	3252675	14252341	54693803	91.00
	Pat Skottle	5678345	65647278	59337698	99.00

-- STORED PROCEDURE

```
DELIMITER //
CREATE PROCEDURE FindMySchedule(IN firstName VARCHAR(45), lastName
VARCHAR(45))
BEGIN
    SELECT StudentID, SectionID, CourseName, CourseCredit, SectionDays, SectionTime,
RoomNumber
    FROM Student JOIN SectionStudent USING(StudentID) JOIN Section USING(SectionID)
JOIN Course USING(CourseID)
    WHERE StudentFName LIKE firstName AND StudentLName LIKE lastName;
END //
DELIMITER ;
CALL FindMySchedule('Avis', 'Denziloe');
```

--

	StudentID	SectionID	CourseName	CourseCredit	SectionDays	SectionTime	RoomNumber
▶	93714844	17009066	GALA1474	2	TTH	10:55 AM	A209