CANVAS

SUMMARY REPORT

Group 9
Ghasmin Hendereldy Guillen Postigo
Tom Kingston
Rahulgopal Sarvisetty
Sai Eshwar Tadepalli

IS 6420 Spring 2023

TABLE OF CONTENT

Executive Summary

The Canvas student app is a vital tool created to improve both students and teachers educational experiences. Our software gives students the ability to manage their school work and engage with peers, while also giving teachers the tools they need to deliver lessons more efficiently. It does this by combining an easy-to-use interface with a rich feature set.

The app's main features include letting users access and manage their courses, including all assignments, deadlines, and course materials. Using group projects and discussion boards, students can work with peers to make learning more interesting and immersive. Additionally, the app offers students immediate and customized feedback on their assignments and tests, assisting them in staying on track and maximizing their potential. Last but not least, the app is completely mobile device friendly, enabling students to access their schoolwork and engage with peers whenever and wherever they choose.

One of the advantages of the Canvas application is that it encourages students to become more involved in their education, which has been proven to enhance academic results. The software streamlines course management and assessment procedures, giving instructors more time to concentrate on engaging students and instructing. Also, the app encourages student cooperation and discussion, resulting in a more interesting and immersive learning experience. The app's accessibility from anywhere, at any time, enables students to study and work together on their own timetable.

For students and teachers who want to enhance their educational experience, the Canvas student app is a crucial tool. The software enables students to be more engaged in their education and instructors to deliver lessons more successfully because of its extensive feature set, user-friendly layout, and mobile compatibility. The application helps to increase student results and promote lifelong learning by encouraging collaboration, engagement, and feedback.

Transactional Database

Transactional databases are essential to the operation of the student canvas app because they offer a consistent and effective method of managing the complex data related to coursework, student profiles, and collaborative activities.

To store and handle all of the data relating to student profiles, course enrollment, assignments, grades, and discussions, the student canvas app makes use of a transactional database. The database keeps track of every transaction, including the addition of a new course and the submission of an assignment, along with key information like the date, time, and user ID. This enables the app to keep track of students' progress and performance in real-time and to offer them individualized feedback and suggestions depending on their behavior and performance.

The student canvas app's transactional database facilitates student cooperation and conversation by enabling them to exchange ideas, criticism, and resources in real-time. Each group project and discussion post is saved in the database along with important information such the user IDs, timestamps, and content. This makes it possible for students to follow the development of their group projects and participate in a fun and engaging way with their classmates.

All things considered, the transactional database employed by the student canvas app is a crucial tool for organizing the intricate data related to coursework, student profiles, and collaborative tasks. This database offers a dependable and effective way to support the main functions of the app by enforcing stringent rules regarding how data is stored, updated, and accessed. This enables students to learn and collaborate more effectively and instructors to offer more individualized feedback and support.

GENERAL DESCRIPTION

BACKGROUND & HISTORY

Instructure, the creator of Canvas, was founded in 2008 by two graduate students from Brigham Young University, Brian Whitmer and Devlin Daley (Instructure, 2018). In 2011, Canvas was launched as a learning management system to allow students to access their courses and groups using a mobile device. Through this app students and teachers can submit assignments, participate in discussions, view grades and course materials. The app also provides access to course calendars, To Do items, notifications, and Conversations messages.

Here are some quick facts from the company's website (Instructure, 2023):

- Founded in 2008
- Launched Canvas in 2011
- Launched Canvas Network in Canvas Network began offering open, online courses for everyone, everywhere in 2012
- Instructure went public on the NYSE (INST) in 2015
- Canvas reached more than 30 million global users. The online Canvas community surpassed 500,000 users. And Instructure acquired Portfolium and MasteryConnect in 2019

VISION AND GOALS

Their vision is to inspire everyone to learn together. 3 Their goals are focused on education impact, employee development, community engagement, flexibility, company culture and opportunity.

PRIORITY REQUIREMENT SUMMARY

For the scope of this project, since the application has different roles such as teacher, administrator, student, etc. we will be focused on the student perspective. The following features are the most important to the student experience:

- Students must be able to see a calendar with assignments and quizzes' due dates for each course they are registered for.
- Courses should be available for the students with discussion boards and grades.
- Each course will have an accumulated grade of quizzes and assignments.
- Students can create or answer comments, send messages in every course discussion board
- Students can see the history of their recent visits to each assignment, test, or announcement, with dates and courses
- Students will see a reminder for missed quizzes and assignments

Based on our team's analysis of these requirements, we created the following tables

- Student
- Course
- Teacher
- Register
- Grade
- Assignment
- Document type
- Quiz
- Question type

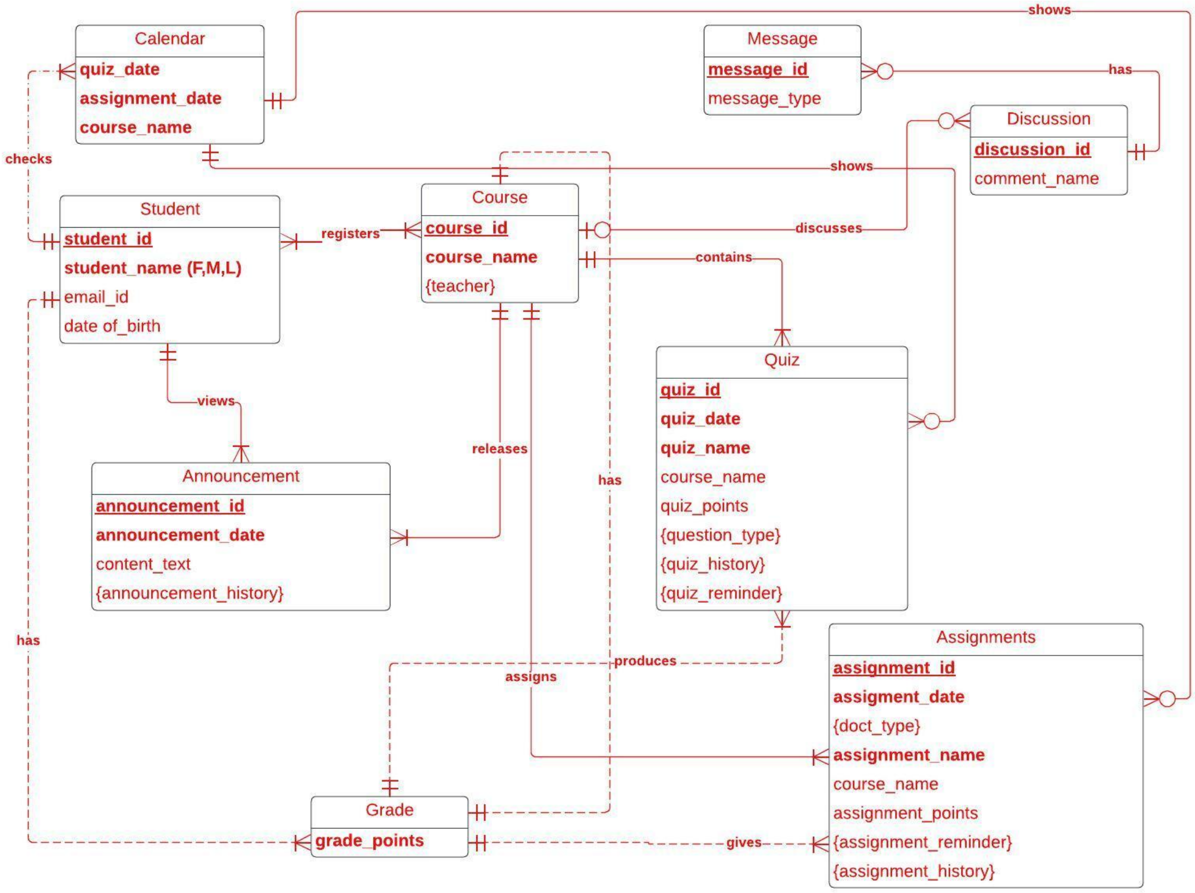
In order to fulfill the requirements, the mentioned tables will record student information: student ID, name, email, date of birth. In addition, information about every course: course ID, teacher, grades, names and due dates for every assignment and quiz. Our team has created the tables and inserted data using Structured Query Language SQL. So far the database contains: 10 students, 10 courses, 10 quizzes, 10 assignments. Moreover, our team will suggest future features in user requirements.

Ethical Considerations

The implementation of a database for Canvas raises ethical considerations that developers must consider. The collection, analysis, and distribution of data must be done with utmost care to protect the privacy of students and faculty. Security measures such as encryption, access controls, and monitoring of user activity are necessary to prevent unauthorized access to the data.

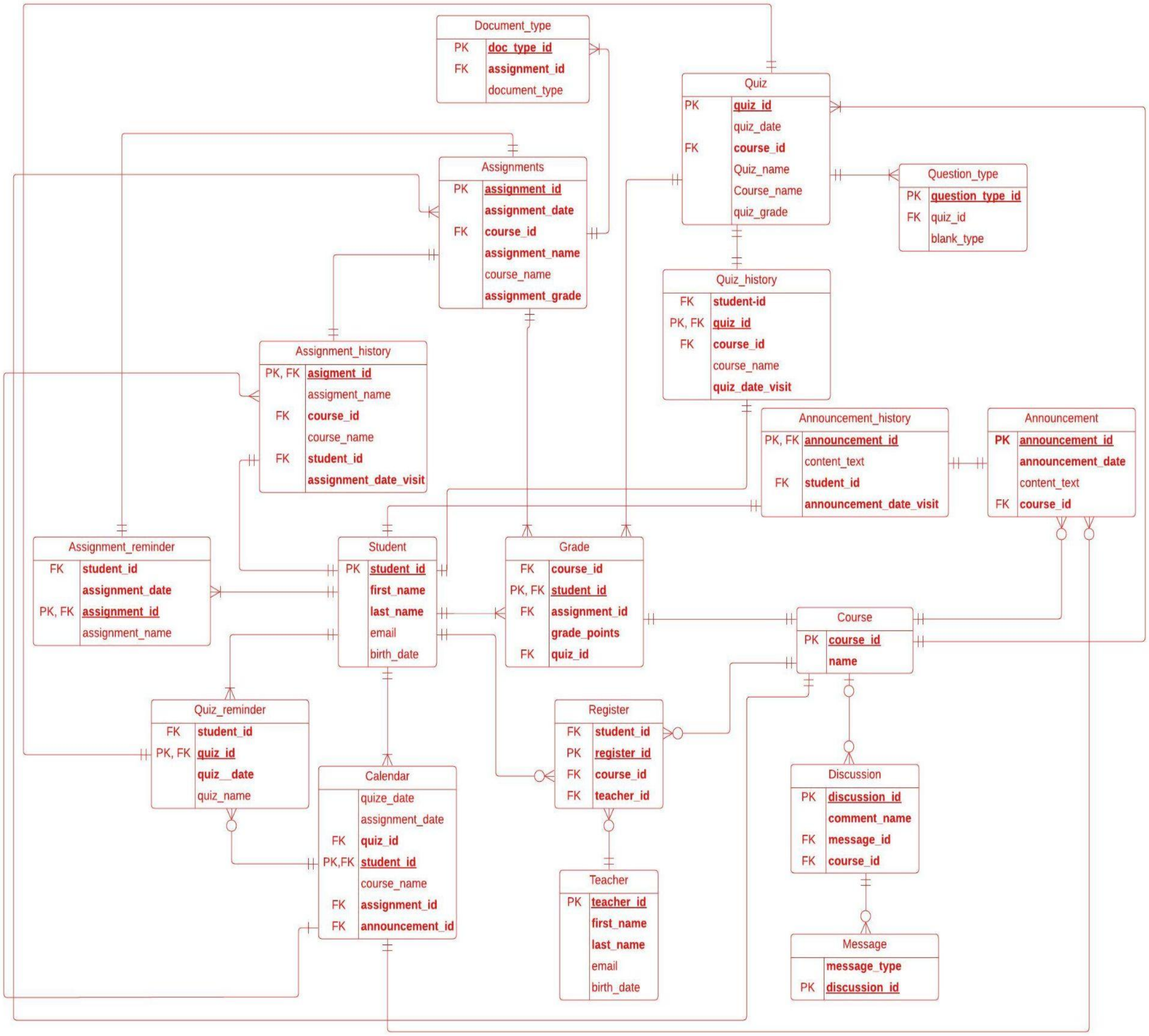
Transparency and informed consent are also important ethical considerations, and Canvas developers should communicate their data collection policies clearly to users. Fairness is another consideration, and developers must ensure that the data is accurate and unbiased, and that decision-making based on the data is transparent and fair. In summary, Canvas developers must implement security and oversight measures to responsibly manage the dataset, while also considering ethical considerations such as privacy, transparency, fairness, and consent.

Conceptual Model



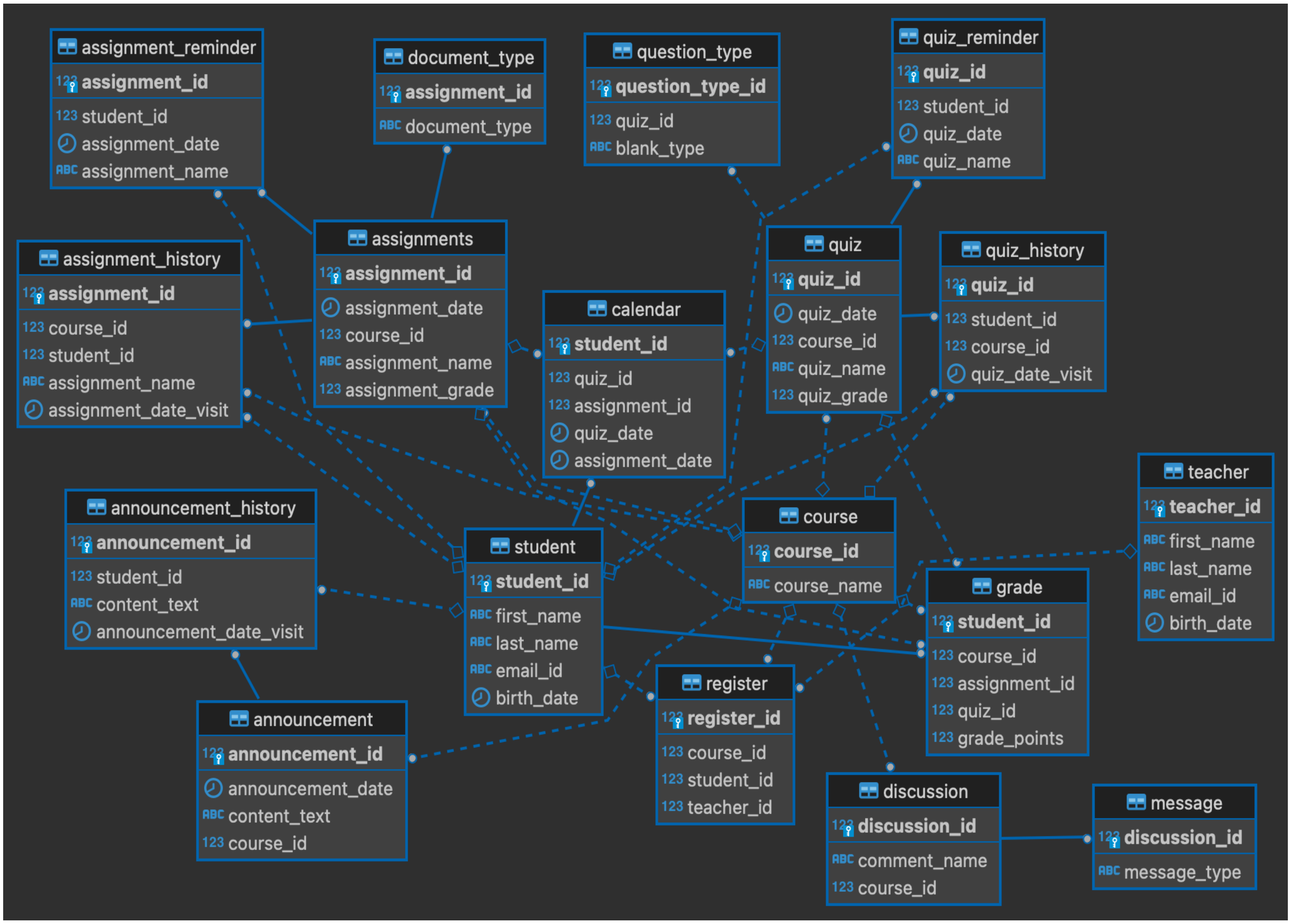
Making a conceptual model is the initial step in the database building process. The conceptual model we developed for Canvas is shown in the diagram above. We construct the model in accordance with the corporate goals on the canvas. The model's focal point is the student table. Each student has a unique ID that contains particular details, including their first and last names and birthdate. The model makes extensive use of constraints of the many-to-many relationships. We have made sure that each student can register for a course and view quizzes and assignments in order to connect the student table to other tables. We linked the quiz and assignment tables to the course table so that students could see the associated information as well as the individual quiz and assignment grades. A grade table that includes the overall grade for each subject was also made.

Logical Model



The project's second stage is: Logical model, which is based on Conceptual modeling with more details. Every table has primary keys and foreign keys correctly identified, binary many-to-many relationships replaced by entity "bridges" connected through foreign keys, such as the register table. We created the following entities based on the multi-valued attributes: Teacher, question type, and document type. This logical model is our program planning, implementation, management, evaluation, and reporting tool.

Physical Model



Canvas is an open-source Learning Management System (LMS) that uses MySQL as its database management system. In MySQL DBeaver, the physical model for Canvas's database schema consists of a set of tables that store information related to various aspects of the LMS. For example, the student table stores information about the students of the system, such as their first name, email, and birthdate. Similarly, the courses table stores information about the courses offered on the platform, such as their course id and course name. Other tables in the schema store information about assignments, quizzes, discussions, and other activities that take place on the platform. The physical model of the Canvas database in MySQL DBeaver provides a clear and structured representation of the data that is stored in the system, making it easy for developers to work with and maintain the system.

Sample Data Used

For the sample data that we used in our database was categorized into the following lists:

- 10 Students
- 10 Courses
- 10 Teachers
- 10 Quizzes
- 10 Assignments

For Students and Courses we used a web generator to generate the random data according to the data type. The grades were created for each student by choosing the grades from individual assignments and quizzes based on the course type.

Requirement Review

S.No	Requirements	Status
1	Students must be able to see a calendar with assignment and quiz due dates for each course they are registered for.	**Completed**
2	Courses should be available for the students with discussion and grade.	**Completed**
3	Each course will have an accumulated grade of quizzes and assignments.	**Completed**
4	Students can create or answer comments, send messages in every course discussion.	**Completed**
5	Students can see the history of their recent visits to each assignment, quiz, or announcement, with dates and courses.	**Completed**
6	Students will see a reminder for missed quizzes and assignments.	**Completed**
7	Students can seek help from the Canvas administrator using the help option.	**In Future**
8	A feature of video call within canvas, which can be used for conducting online classes.	**In Future**
9	A feature of virtual reality and gamification for making learning activities of students engaging and having fun.	**In Future**

Conclusion - Products/services provided by the organization etc

Canvas is a learning management system (LMS) that provides a range of services to facilitate teaching and learning. It is used by millions of users worldwide and is a popular choice for higher education institutions. One of the primary services Canvas provides is course management. Educators can use Canvas to create and manage course content, including assignments, quizzes, and discussions. Canvas also provides tools for grading and assessment, allowing educators to provide feedback and monitor student progress. Students can access course materials and complete assignments through Canvas, making it easier to stay organized and on track with their coursework.

Another service Canvas provides is communication tools. Educators and students can communicate through Canvas using messaging, discussions, and announcements. This allows for easy communication between teachers and students and promotes collaboration and engagement. Canvas also provides tools for conferencing and video recording, making it easy for educators to conduct virtual classes or meetings. Canvas also provides services for data analytics and reporting. Educators can use Canvas to track student progress, monitor course outcomes, and analyze data to improve teaching and learning. Canvas provides detailed analytics and reporting tools that allow educators to track student engagement, identify areas of improvement, and make data-driven decisions.

One of the key benefits of Canvas is its flexibility and accessibility. Canvas can be accessed from any device with an internet connection, making it easy for students to stay connected and engaged with their coursework. Canvas also provides support for accessibility, making it easier for students with disabilities to access and engage with course content. In addition to these services, Canvas also provides integration with third-party tools and services. This allows educators to integrate Canvas with other tools they use in their teaching, such as plagiarism checkers, digital textbooks, and multimedia tools.

In conclusion, Canvas provides a range of services that can benefit both students and educators. Course management, communication tools, data analytics, and reporting are just a few of the services that Canvas provides. Its flexibility and accessibility make it a popular choice for educators and students alike, and its integration with third-party tools allows for even more customization and flexibility.

References

- Sample Data for Students and Teachers Name has been generated from Mockaroo website.

- Prusinski, Wojciech, et al. "Designing Efficient and Scalable Databases for Big Data Analytics." Journal of Big Data, vol. 7, no. 1, 2020, pp. 1-28. doi: 10.1186/s40537-020-00307-6.

- Garcia-Molina, Hector, et al. Database Systems: The Complete Book. 2nd ed., Pearson, 2009.

- Elmasri, Ramez, et al. Fundamentals of Database Systems. 7th ed., Pearson, 2016.

- Silberschatz, Abraham, et al. Database System Concepts. 6th ed., McGraw-Hill, 2010.

- Connolly, Thomas M., and Carolyn E. Begg. Database Systems: A Practical Approach to Design, Implementation, and Management. 5th ed., Pearson, 2015.

- Ramakrishnan, Raghu, and Johannes Gehrke. Database Management Systems. 3rd ed., McGraw-Hill, 2003.

- Date, C. J. An Introduction to Database Systems. 8th ed., Addison-Wesley, 2003.

- Navathe, Shamkant B., et al. Fundamentals of Database Systems. 7th ed., Pearson, 2015.

Appendix

Ghasmin Hendereldy Guillen Postigo - 19 hours

Group Meetings: 12 hours

- Logical Diagram 2 hours
- Physical Diagram 4 hours
- Create Database 4 hours
- Compile Information 2 hours

Write Vision and Objectives of Canvas: 1 hour
Write Priority Requirements: 2 hours
Compile Final Presentation: 4 hours

Tom Kingston - 19 hours

Group Meetings: 12 hours

- Logical Diagram 2 hours
- Physical Diagram 4 hours
- Create Database 4 hours
- Compile Information 2 hours

Write Ethical Considerations: 1 hour
Write Product/services provided by Canvas: 2 hours
Compile Final Paper: 4 hours

Rahulgopal Sarvisetty - 19 hours

Group Meetings: 12 hours

- Logical Diagram 2 hours
- Physical Diagram 4 hours
- Create Database 4 hours
- Compile Information 2 hours

Write Requirement Review: 3 hour
Compile Database coding and add data: 4 hours

Saieshwar Tadepalli - 19 hours

Group Meetings: 12 hours

- Logical Diagram 2 hours
- Physical Diagram 4 hours
- Create Database 4 hours
- Compile Information 2 hours

Write Executive Summary: 1 hour
Write Description of how transactional databases are used to support operations: 2 hours
Compile Database coding and add data: 4 hours

SQL Statements - Create Tables

```
DROP TABLE IF EXISTS Student;
CREATE TABLE Student(
     student_id INTEGER NOT NULL,
     first_name VARCHAR (35) NOT NULL,
     last_name VARCHAR (35) NOT NULL,
     email_id VARCHAR (200),
     birth_date DATE,
     PRIMARY KEY (student_id)
);

DROP TABLE IF EXISTS Course;
CREATE TABLE Course(
     course_id INTEGER NOT NULL,
     course_name VARCHAR(255) NULL NULL,
     PRIMARY KEY (course_id)
);

DROP TABLE IF EXISTS Teacher;
CREATE TABLE Teacher(
     teacher_id INTEGER NOT NULL,
     first_name VARCHAR (35) NOT NULL,
     last_name varchar (35) NOT NULL,
     email_id VARCHAR (200),
     birth_date DATE,
     PRIMARY KEY (teacher_id)
);

DROP TABLE IF EXISTS Register;
CREATE TABLE Register(
register_id integer not null,
course_id INTEGER,
student_id INTEGER,
teacher_id Integer,
primary KEY (register_id),
CONSTRAINT register_fk_course_id foreign KEY (course_id) REFERENCES
Course (course_id),
CONSTRAINT register_fk_student_id foreign KEY (student_id) REFERENCES
Student (student_id),
CONSTRAINT register_fk_teacher_id foreign KEY (teacher_id) REFERENCES
Teacher (teacher_id)
);

DROP TABLE IF EXISTS Assignments;
CREATE TABLE Assignments(
     assignment_id INTEGER NOT NULL,
     assignment_date DATE NOT NULL,
     course_id INTEGER,
     assignment_name VARCHAR(255) NOT NULL,
     assignment_grade float(20),
     PRIMARY KEY(assignment_id),
     CONSTRAINT course_fKEY_course_id FOREIGN KEY (course_id)
REFERENCES Course (course_id)
);

DROP TABLE IF EXISTS Document_type;
CREATE TABLE Document_type(
     assignment_id INTEGER NOT null,
     PRIMARY KEY(assignment_id),
     document_type VARCHAR(20),
     CONSTRAINT assignment_fKEY_assignment_id FOREIGN KEY
(assignment_id) REFERENCES Assignments (assignment_id)
);

DROP TABLE IF EXISTS Quiz;
CREATE TABLE Quiz(
     quiz_id INTEGER NOT NULL,
     quiz_date DATE NOT NULL,
     course_id INTEGER,
     quiz_name VARCHAR(255) NOT NULL,
     quiz_grade float(20),
     PRIMARY KEY(quiz_id),
     CONSTRAINT course_fKEY_course_id FOREIGN KEY (course_id)
REFERENCES Course (course_id)
);

DROP TABLE IF EXISTS Question_type;
CREATE TABLE Question_type(
     question_type_id INTEGER NOT null,
     quiz_id INTEGER NOT null,
     blank_type VARCHAR(255),
     PRIMARY KEY(question_type_id),
     CONSTRAINT quiz_fKEY_quiz_id FOREIGN KEY (quiz_id) REFERENCES
Quiz (quiz_id)
);

DROP TABLE IF EXISTS Grade;
CREATE TABLE Grade(
     course_id INTEGER,
     student_id INTEGER NOT NULL,
     assignment_id INTEGER,
     quiz_id INTEGER,
     grade_points FLOAT NOT NULL,
     PRIMARY KEY(student_id),
     CONSTRAINT course_fKEY_course_id FOREIGN KEY (course_id)
REFERENCES Course (course_id),
     CONSTRAINT student_fKEY_student_id FOREIGN KEY (student_id)
REFERENCES Student (student_id),
     CONSTRAINT assignment_fKEY_assignment_id FOREIGN KEY
(assignment_id) REFERENCES Assignments (assignment_id),
     CONSTRAINT quiz_fKEY_quiz_id FOREIGN KEY (quiz_id) REFERENCES
Quiz (quiz_id)
);

DROP TABLE IF EXISTS Discussion;
CREATE TABLE Discussion(
     discussion_id INTEGER NOT NULL,
     comment_name VARCHAR(255),
     course_id INTEGER,
     PRIMARY KEY (discussion_id),
     CONSTRAINT dis_fKEY_course_id FOREIGN KEY (course_id) REFERENCES
Course (course_id)
);

DROP TABLE IF EXISTS Message;
CREATE TABLE Message(
message_type VARCHAR(255),
discussion_id INTEGER NOT null,
CONSTRAINT mes_fKEY_discussion_id foreign KEY (discussion_id)
REFERENCES Discussion (discussion_id),
primary KEY (discussion_id)
```

```sql
);

DROP TABLE IF EXISTS Quiz_Reminder;
CREATE TABLE Quiz_Reminder(
     student_id INTEGER,
     quiz_id INTEGER NOT NULL,
     quiz_date DATE,
     quiz_name VARCHAR(255),
     PRIMARY KEY (quiz_id),
     CONSTRAINT quiz_rem_fKEY_student_id FOREIGN KEY (student_id) REFERENCES Student (student_id),
     CONSTRAINT quiz_rem_fKEY_quiz_id FOREIGN KEY (quiz_id) REFERENCES Quiz (quiz_id)
);

DROP TABLE IF EXISTS Assignment_Reminder;
CREATE TABLE Assignment_Reminder(
     student_id INTEGER,
     assignment_id INTEGER NOT NULL,
     assignment_date DATE,
     assignment_name VARCHAR(255),
     PRIMARY KEY (assignment_id),
     CONSTRAINT assign_rem_fKEY_student_id foreign KEY (student_id) REFERENCES Student (student_id),
     CONSTRAINT assign_rem_fKEY_assignment_id foreign KEY (assignment_id) REFERENCES Assignments (assignment_id)
);

DROP TABLE IF EXISTS Announcement;
CREATE TABLE Announcement(
     announcement_id INTEGER NOT NULL,
     announcement_date DATE,
     content_text VARCHAR(255),
     course_id integer,
     PRIMARY KEY (announcement_id),
     CONSTRAINT announcement_fKEY_course_id foreign KEY (course_id) REFERENCES Course (course_id)
);

DROP TABLE IF EXISTS Announcement_history;
CREATE TABLE Announcement_history(
     announcement_id INTEGER NOT NULL,
     student_id INTEGER,
     content_text varchar(255),
     announcement_date_visit date,
     PRIMARY KEY (announcement_id),
     CONSTRAINT anh_fKEY_announcement_id FOREIGN KEY (announcement_id)
REFERENCES Announcement (announcement_id),
     CONSTRAINT anh_fKEY_student_id FOREIGN KEY (student_id)
REFERENCES Student (student_id)
);

DROP TABLE IF EXISTS Assignment_history;
CREATE TABLE Assignment_history(
assignment_id INTEGER NOT NULL,
course_id INTEGER,
student_id INTEGER,
assignment_name VARCHAR(255),
assignment_date_visit date,
PRIMARY KEY (assignment_id),
CONSTRAINT ah_fKEY_assignment_id FOREIGN KEY (assignment_id)
REFERENCES Assignments (assignment_id),
CONSTRAINT ah_fKEY_student_id FOREIGN KEY (student_id) REFERENCES
Student (student_id),
CONSTRAINT ah_fKEY_course_id FOREIGN KEY (course_id) REFERENCES Course
(course_id)
);

DROP TABLE IF EXISTS Quiz_history;
CREATE TABLE Quiz_history(
student_id INTEGER,
course_id INTEGER,
quiz_id INTEGER NOT null,
quiz_date_visit date,
PRIMARY KEY (quiz_id),
CONSTRAINT qh_fKEY_student_id FOREIGN KEY (student_id) REFERENCES
Student (student_id),
CONSTRAINT qh_fKEY_course_id FOREIGN KEY (course_id) REFERENCES Course
(course_id),
CONSTRAINT ah_fKEY_quiz_id FOREIGN KEY (quiz_id) REFERENCES Quiz
(quiz_id)
);
```

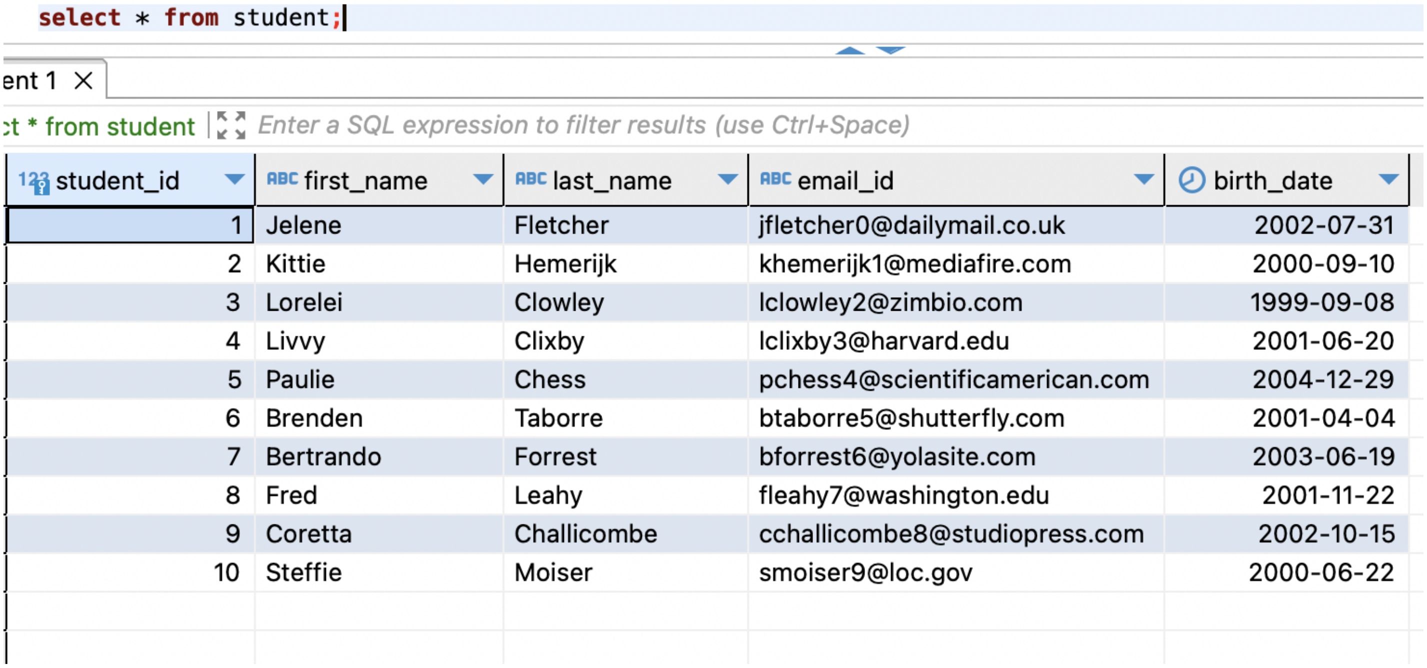
```
DROP TABLE IF EXISTS Calendar;
CREATE TABLE Calendar(
	quiz_id INTEGER,
	assignment_id INTEGER,
	student_id INTEGER NOT NULL,
	quiz_date DATE NOT NULL,
	assignment_date DATE NOT NULL,
	PRIMARY KEY (student_id),
	CONSTRAINT quiz_fKEY_quiz_id FOREIGN KEY (quiz_id) REFERENCES
Quiz (quiz_id),
	CONSTRAINT assignment_fKEY_assignment_id FOREIGN KEY
(assignment_id) REFERENCES Assignments (assignment_id),
	CONSTRAINT student_fKEY_student_id FOREIGN KEY (student_id)
REFERENCES Student (student_id)
);
```

Sample Insert Statements

```
-- Student table

INSERT INTO Student (student_id, first_name, last_name, email_id,
birth_date) VALUES
(1, 'Jelene', 'Fletcher', 'jfletcher0@dailymail.co.uk', '2002-07-31'),
(2, 'Kittie', 'Hemerijk', 'khemerijk1@mediafire.com', '2000-09-10'),
(3, 'Lorelei', 'Clowley', 'lclowley2@zimbio.com', '1999-09-08'),
(4, 'Livvy', 'Clixby', 'lclixby3@harvard.edu', '2001-06-20'),
(5, 'Paulie', 'Chess', 'pchess4@scientificamerican.com',
'2004-12-29'),
(6, 'Brenden', 'Taborre', 'btaborre5@shutterfly.com', '2001-04-04'),
(7, 'Bertrando', 'Forrest', 'bforrest6@yolasite.com', '2003-06-19'),
(8, 'Fred', 'Leahy', 'fleahy7@washington.edu', '2001-11-22'),
(9, 'Coretta', 'Challicombe', 'cchallicombe8@studiopress.com',
'2002-10-15'),
(10, 'Steffie', 'Moiser', 'smoiser9@loc.gov', '2000-06-22');

-- Course table

INSERT INTO Course (course_id, course_name) VALUES
(1, 'Introduction to Business Analytics'),
(2, 'Database Theory and Design'),
(3, 'Data Structures and Java'),
(4, 'Artificial Intelligence'),
(5, 'Software Engineering'),
(6, 'Business Development'),
(7, 'Cyber Security'),
(8, 'Statistics and Predictive Analysis'),
(9, 'Data Engineering'),
(10, 'Data Visualization');

-- Teacher table

INSERT INTO Teacher (teacher_id, first_name, last_name, email_id,
birth_date) VALUES
(1, 'Dr. Jane', 'Smith', 'jane.smith@example.com', '1975-01-01'),
(2, 'Prof. John', 'Brown', 'john.brown@example.com', '1980-02-02'),
(3, 'Dr. Mary', 'Davis', 'mary.davis@example.com', '1985-03-03'),
(4, 'Prof. William', 'Johnson', 'william.johnson@example.com',
'1990-04-04'),
(5, 'Dr. Elizabeth', 'Wilson', 'elizabeth.wilson@example.com',
'1995-05-05'),
(6, 'Prof. Julio', 'Leftwich', 'jleftwichl@chicagotribune.com',
'1960-06-05'),
(7, 'Dr. Gabrielle', 'Kemme', 'gkemmem@vk.com', '1970-05-30'),
(8, 'Dr. Gabrielle', 'Tonkes', 'ltonkesn@slideshare.net',
'1982-11-22'),
(9, 'Prof. Aron', 'Larrosa', 'alarrosao@nba.com', '1958-08-11'),
(10, 'Prof. Lindy', 'Hoggins', 'lhogginsp@nba.com', '1965-10-10');

-- Register table

INSERT INTO Register (register_id, course_id, student_id, teacher_id)
VALUES
(1, 1, 1, 1),
(2, 2, 1, 2),
(3, 2, 2, 2),
(4, 3, 3, 3),
(5, 4, 4, 4),
(6, 5, 5, 5),
(7, 6, 6, 6),
(8, 7, 7, 7),
(9, 8, 8, 8),
(10, 9, 9, 9),
(11, 10, 10, 10),
(12, 1, 10, 1),
```

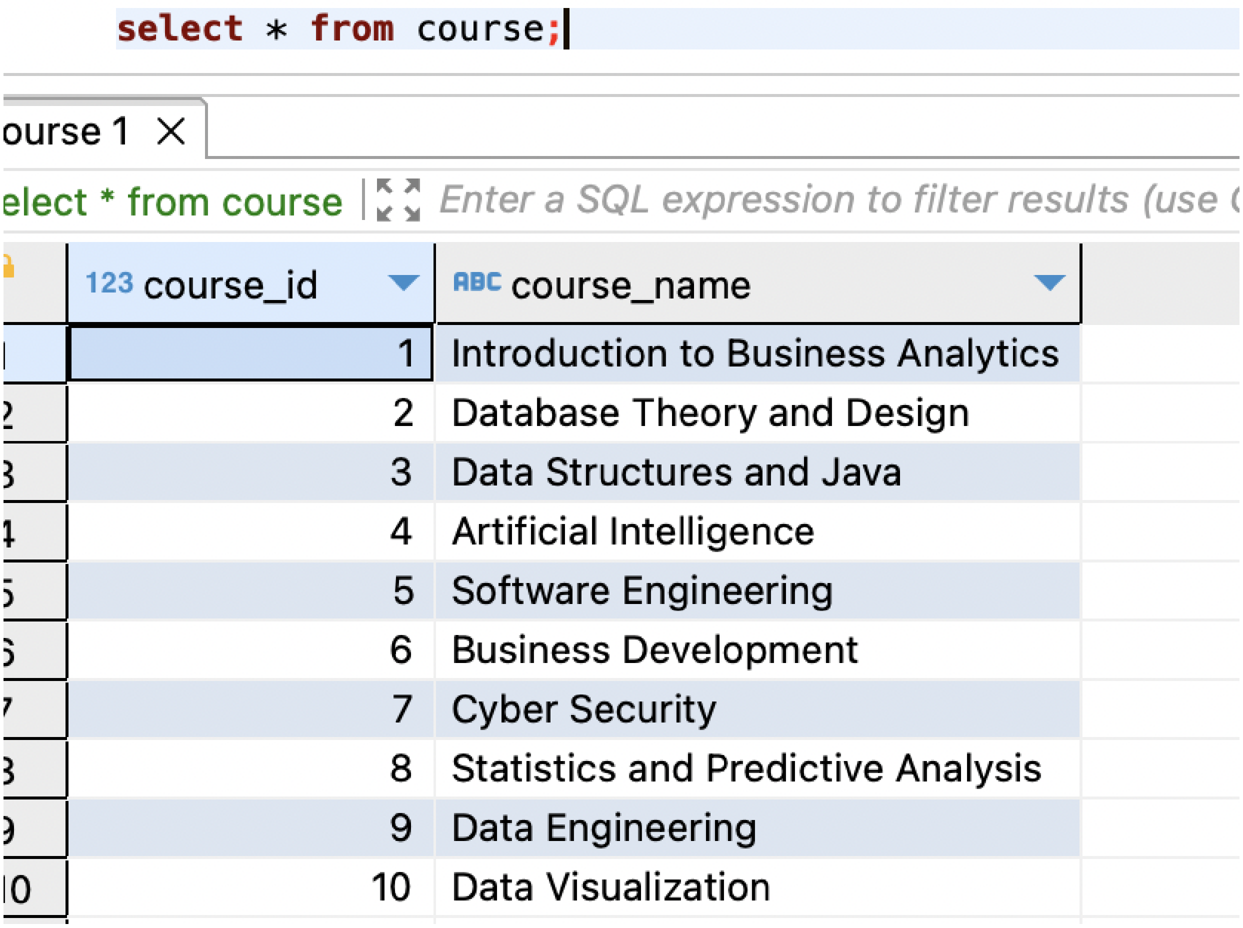
```sql
(13, 5, 6, 5),
(14, 4, 7, 4),
(15, 3, 9, 3);

-- Assignments table

INSERT INTO Assignments (assignment_id, assignment_date, course_id,
assignment_name, assignment_grade) VALUES
(1, '2023-03-01', 1, 'Assignment 1', 3.8),
(2, '2023-03-05', 2, 'Assignment 1', 4),
(3, '2023-03-10', 3, 'Assignment 1', 3.2),
(4, '2023-03-15', 4, 'Assignment 1', 3.94),
(5, '2023-03-20', 5, 'Assignment 1', 3.5),
(6, '2023-03-25', 6, 'Assignment 1', 3.8),
(7, '2023-03-30', 7, 'Assignment 1', 3.5),
(8, '2023-03-31', 8, 'Assignment 2', 4),
(9, '2023-04-05', 9, 'Assignment 2', 3.94),
(10, '2023-04-10', 10, 'Assignment 2', 3.2);

-- Document_type table

INSERT INTO Document_type (assignment_id, document_type) VALUES
(1, 'pdf'),
(2, 'doc'),
(3, 'pdf'),
(4, 'pdf'),
(5, 'doc'),
(6, 'doc'),
(7, 'pdf'),
(8, 'doc'),
(9, 'html'),
(10, 'html');

-- Quiz Table

INSERT INTO Quiz (quiz_id, quiz_date, course_id, quiz_name,
quiz_grade) values
(1,'2023-03-02',1,'Introduction to Business Analytics Quiz 2', 3.4),
(2,'2023-03-04',2,'Database Theory and Design Quiz 1', 3.5),
(3,'2023-03-09',3,'Data Structures and Java Quiz 1', 3.6),
(4,'2023-03-14',4,'Artificial Intelligence Quiz 2', 4),
(5,'2023-03-19',5,'Software Engineering Quiz 3', 3.8),
(6,'2023-04-01',5,'Software Engineering Quiz 3', 3.75),
(7,'2023-03-29',5,'Software Engineering Quiz 3', 3.94),
(8,'2023-03-21',5,'Software Engineering Quiz 3', 4),
(9,'2023-04-06',5,'Software Engineering Quiz 3', 3.8),
(10,'2023-04-09',5,'Software Engineering Quiz 3', 3.5);
```

Sample Insert Statements for Tables



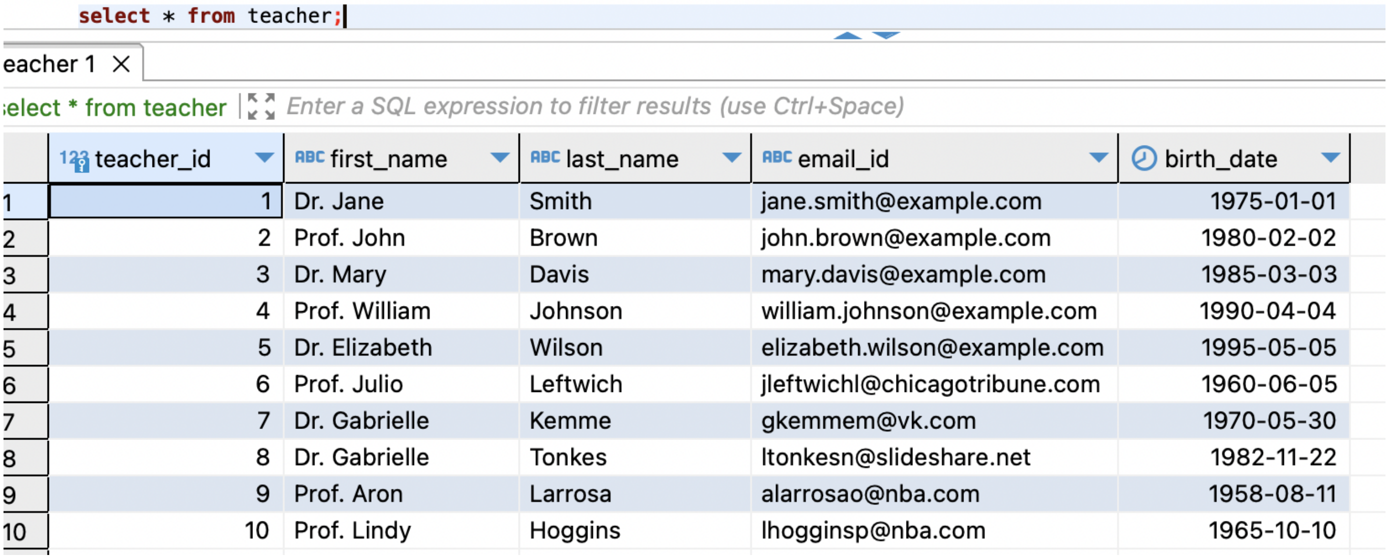
```
select * from student;
```

student_id	first_name	last_name	email_id	birth_date
1	Jelene	Fletcher	jfletcher0@dailymail.co.uk	2002-07-31
2	Kittie	Hemerijk	khemerijk1@mediafire.com	2000-09-10
3	Lorelei	Clowley	lclowley2@zimbio.com	1999-09-08
4	Livvy	Clixby	lclixby3@harvard.edu	2001-06-20
5	Paulie	Chess	pchess4@scientificamerican.com	2004-12-29
6	Brenden	Taborre	btaborre5@shutterfly.com	2001-04-04
7	Bertrando	Forrest	bforrest6@yolasite.com	2003-06-19
8	Fred	Leahy	fleahy7@washington.edu	2001-11-22
9	Coretta	Challicombe	cchallicombe8@studiopress.com	2002-10-15
10	Steffie	Moiser	smoiser9@loc.gov	2000-06-22



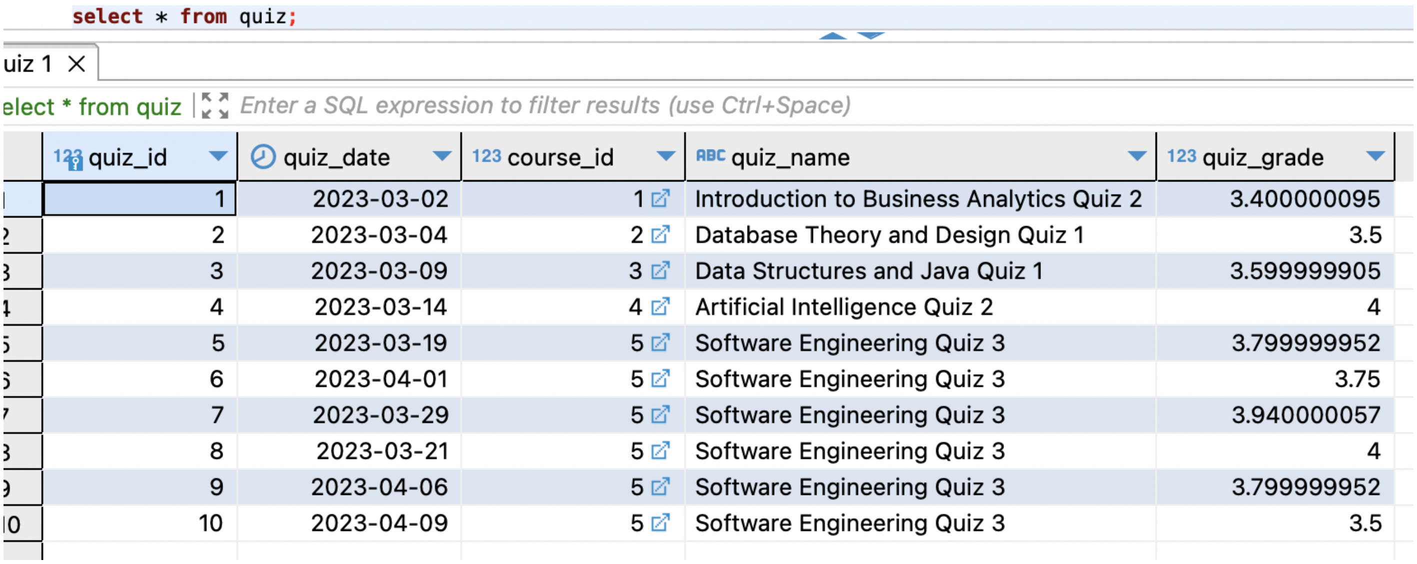
```
select * from course;
```

	course_id	course_name
1	1	Introduction to Business Analytics
2	2	Database Theory and Design
3	3	Data Structures and Java
4	4	Artificial Intelligence
5	5	Software Engineering
6	6	Business Development
7	7	Cyber Security
8	8	Statistics and Predictive Analysis
9	9	Data Engineering
10	10	Data Visualization



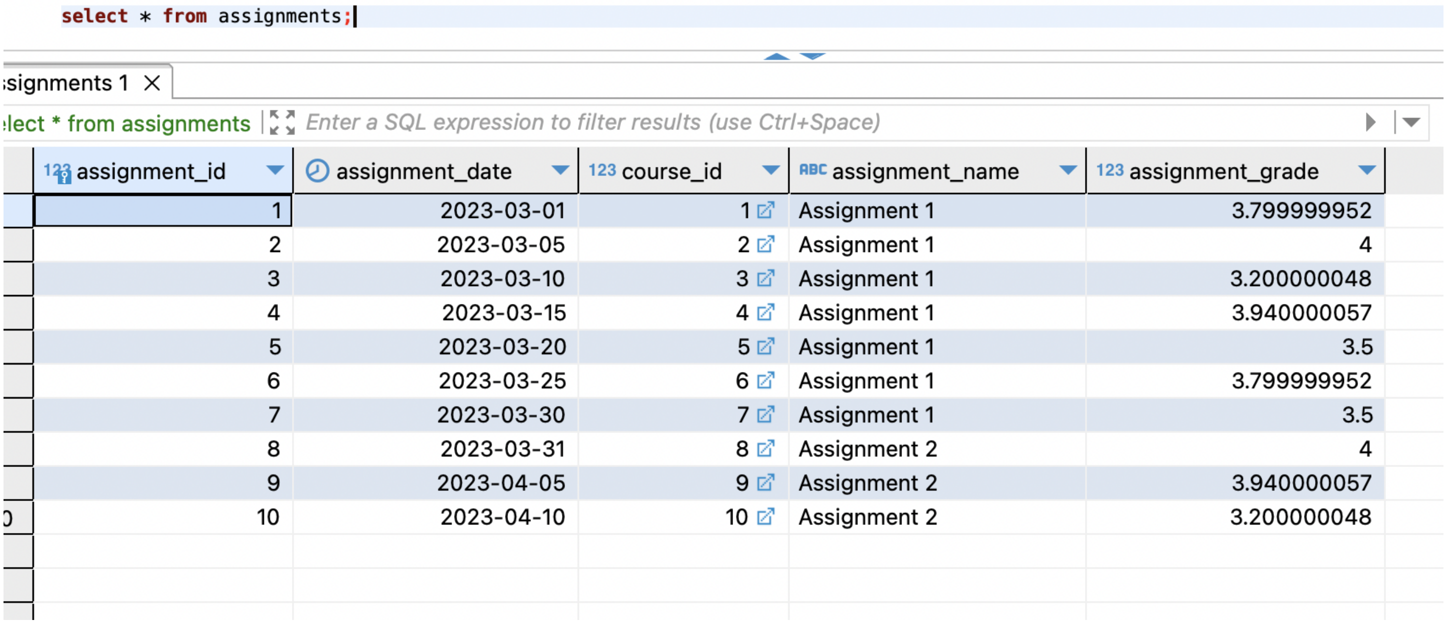
```
select * from teacher;
```

	teacher_id	first_name	last_name	email_id	birth_date
1	1	Dr. Jane	Smith	jane.smith@example.com	1975-01-01
2	2	Prof. John	Brown	john.brown@example.com	1980-02-02
3	3	Dr. Mary	Davis	mary.davis@example.com	1985-03-03
4	4	Prof. William	Johnson	william.johnson@example.com	1990-04-04
5	5	Dr. Elizabeth	Wilson	elizabeth.wilson@example.com	1995-05-05
6	6	Prof. Julio	Leftwich	jleftwichl@chicagotribune.com	1960-06-05
7	7	Dr. Gabrielle	Kemme	gkemmem@vk.com	1970-05-30
8	8	Dr. Gabrielle	Tonkes	ltonkesn@slideshare.net	1982-11-22
9	9	Prof. Aron	Larrosa	alarrosao@nba.com	1958-08-11
10	10	Prof. Lindy	Hoggins	lhogginsp@nba.com	1965-10-10



```
select * from quiz;
```

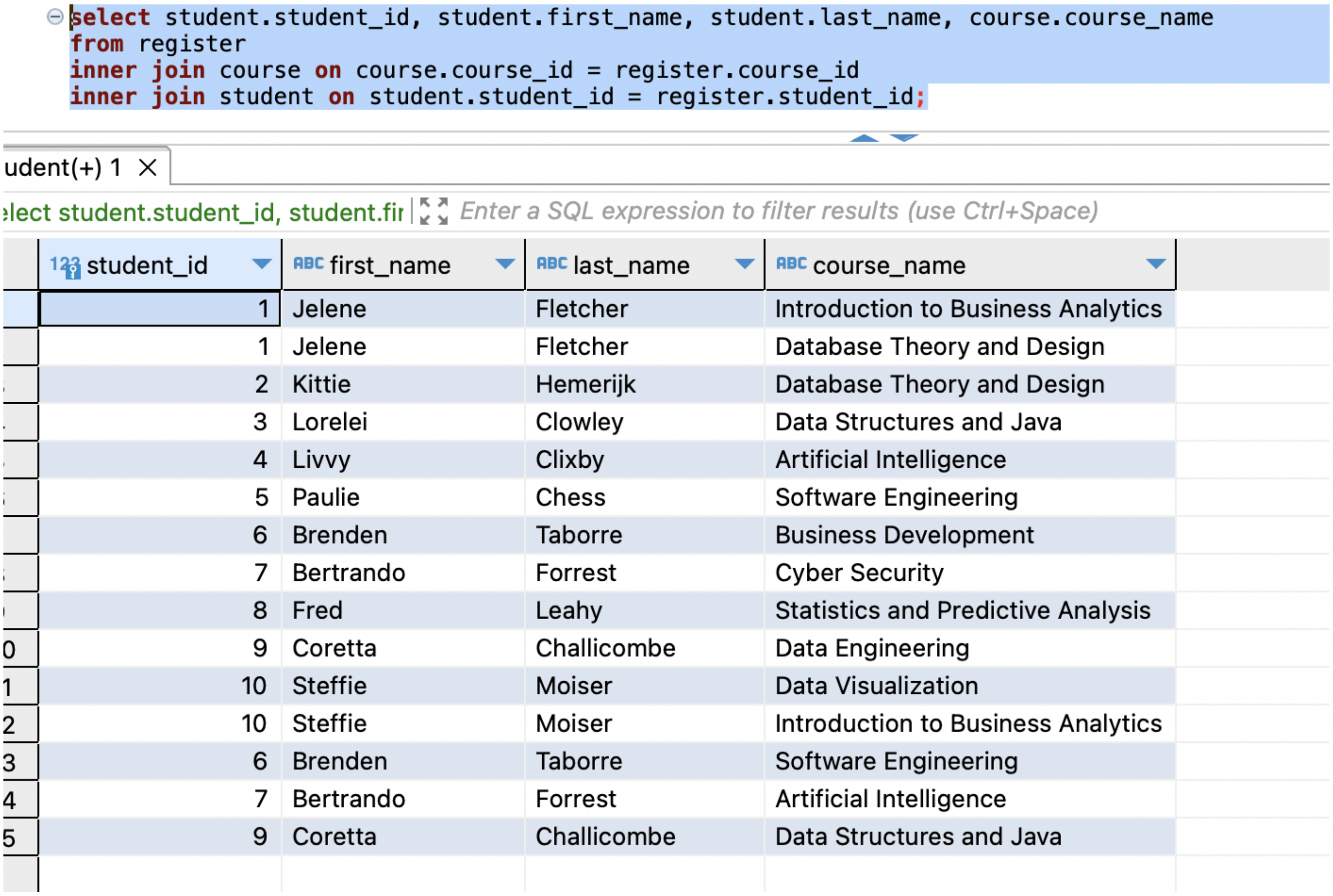
quiz_id	quiz_date	course_id	quiz_name	quiz_grade
1	2023-03-02	1	Introduction to Business Analytics Quiz 2	3.400000095
2	2023-03-04	2	Database Theory and Design Quiz 1	3.5
3	2023-03-09	3	Data Structures and Java Quiz 1	3.599999905
4	2023-03-14	4	Artificial Intelligence Quiz 2	4
5	2023-03-19	5	Software Engineering Quiz 3	3.799999952
6	2023-04-01	5	Software Engineering Quiz 3	3.75
7	2023-03-29	5	Software Engineering Quiz 3	3.940000057
8	2023-03-21	5	Software Engineering Quiz 3	4
9	2023-04-06	5	Software Engineering Quiz 3	3.799999952
10	2023-04-09	5	Software Engineering Quiz 3	3.5



```
select * from assignments;
```

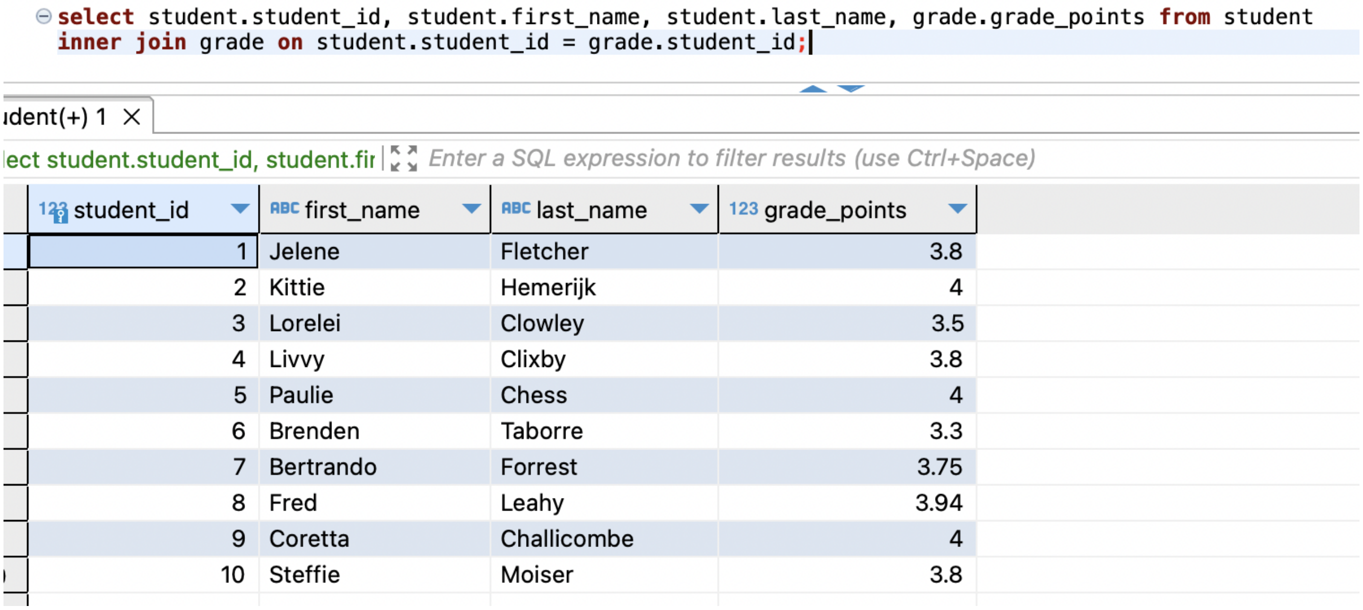
assignment_id	assignment_date	course_id	assignment_name	assignment_grade
1	2023-03-01	1	Assignment 1	3.799999952
2	2023-03-05	2	Assignment 1	4
3	2023-03-10	3	Assignment 1	3.200000048
4	2023-03-15	4	Assignment 1	3.940000057
5	2023-03-20	5	Assignment 1	3.5
6	2023-03-25	6	Assignment 1	3.799999952
7	2023-03-30	7	Assignment 1	3.5
8	2023-03-31	8	Assignment 2	4
9	2023-04-05	9	Assignment 2	3.940000057
10	2023-04-10	10	Assignment 2	3.200000048

Sample Queries


udent(+) 1
ɜlect student.student_id, student.fir
Enter a SQL expression to filter results (use Ctrl+Space)

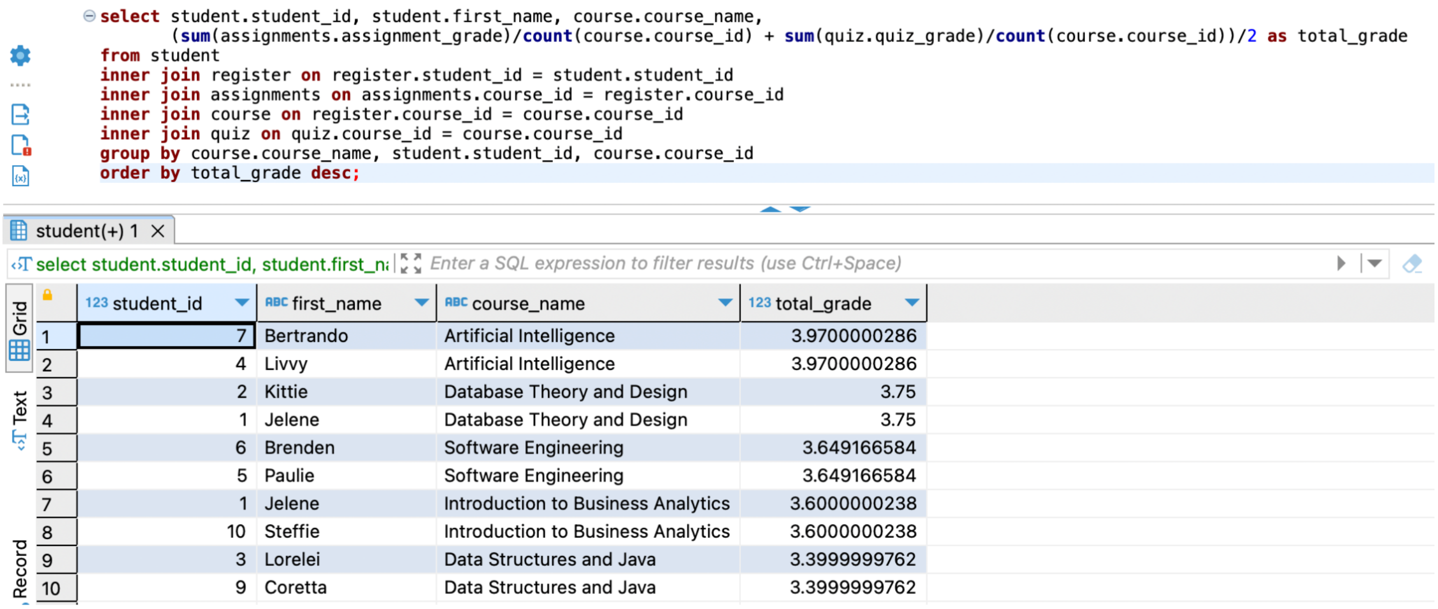
```sql
select student.student_id, student.first_name, student.last_name, course.course_name
from register
inner join course on course.course_id = register.course_id
inner join student on student.student_id = register.student_id;
```

student_id	first_name	last_name	course_name
1	Jelene	Fletcher	Introduction to Business Analytics
1	Jelene	Fletcher	Database Theory and Design
2	Kittie	Hemerijk	Database Theory and Design
3	Lorelei	Clowley	Data Structures and Java
4	Livvy	Clixby	Artificial Intelligence
5	Paulie	Chess	Software Engineering
6	Brenden	Taborre	Business Development
7	Bertrando	Forrest	Cyber Security
8	Fred	Leahy	Statistics and Predictive Analysis
9	Coretta	Challicombe	Data Engineering
10	Steffie	Moiser	Data Visualization
10	Steffie	Moiser	Introduction to Business Analytics
6	Brenden	Taborre	Software Engineering
7	Bertrando	Forrest	Artificial Intelligence
9	Coretta	Challicombe	Data Structures and Java


ıdent(+) 1
ect student.student_id, student.fir
Enter a SQL expression to filter results (use Ctrl+Space)

```sql
select student.student_id, student.first_name, student.last_name, grade.grade_points from student
inner join grade on student.student_id = grade.student_id;
```

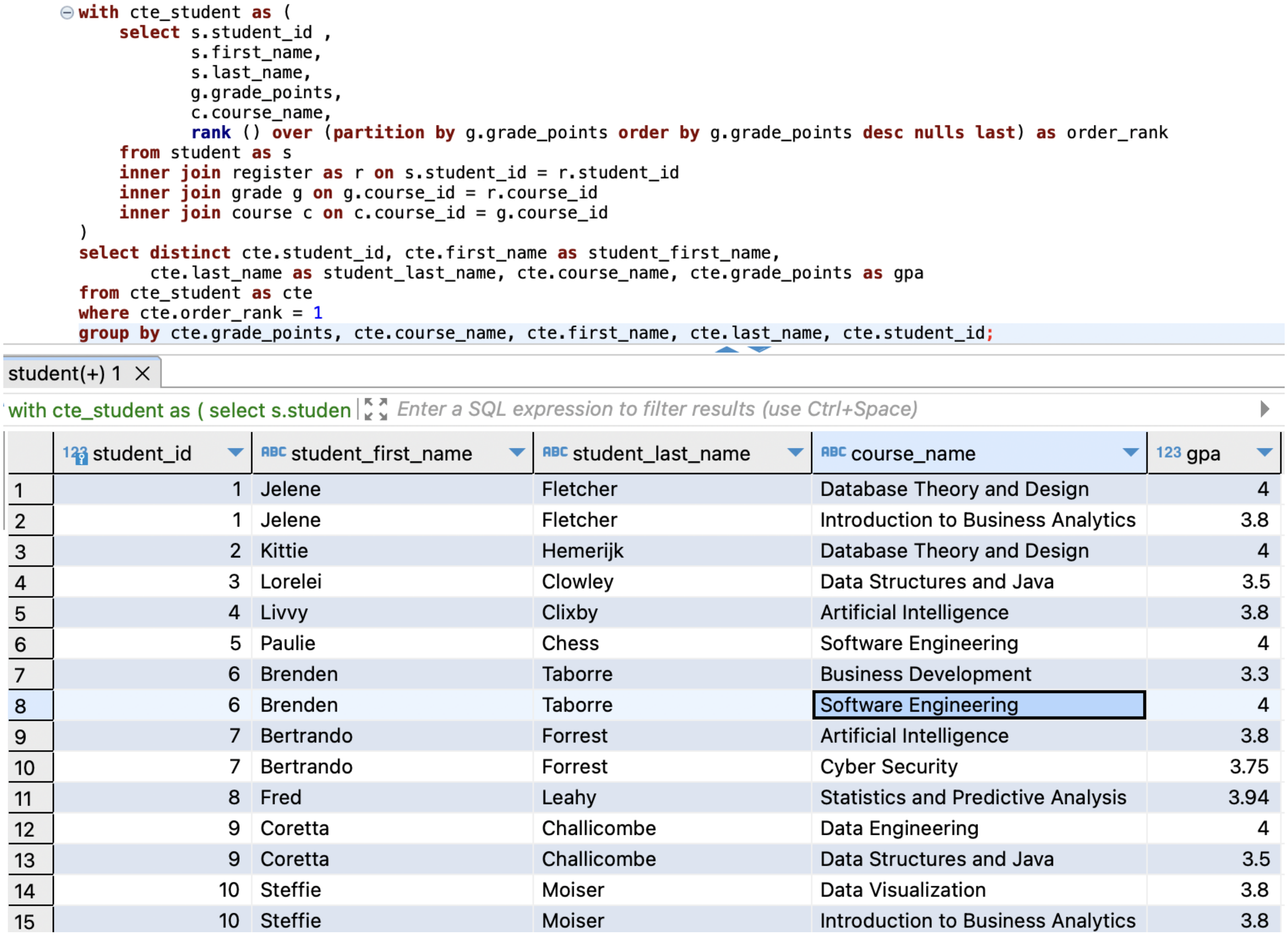
student_id	first_name	last_name	grade_points
1	Jelene	Fletcher	3.8
2	Kittie	Hemerijk	4
3	Lorelei	Clowley	3.5
4	Livvy	Clixby	3.8
5	Paulie	Chess	4
6	Brenden	Taborre	3.3
7	Bertrando	Forrest	3.75
8	Fred	Leahy	3.94
9	Coretta	Challicombe	4
10	Steffie	Moiser	3.8



```
select student.student_id, student.first_name, course.course_name,
       (sum(assignments.assignment_grade)/count(course.course_id) + sum(quiz.quiz_grade)/count(course.course_id))/2 as total_grade
from student
inner join register on register.student_id = student.student_id
inner join assignments on assignments.course_id = register.course_id
inner join course on register.course_id = course.course_id
inner join quiz on quiz.course_id = course.course_id
group by course.course_name, student.student_id, course.course_id
order by total_grade desc;
```

student(+) 1
select student.student_id, student.first_n
Enter a SQL expression to filter results (use Ctrl+Space)
Grid
Text
Record

	student_id	first_name	course_name	total_grade
1	7	Bertrando	Artificial Intelligence	3.9700000286
2	4	Livvy	Artificial Intelligence	3.9700000286
3	2	Kittie	Database Theory and Design	3.75
4	1	Jelene	Database Theory and Design	3.75
5	6	Brenden	Software Engineering	3.649166584
6	5	Paulie	Software Engineering	3.649166584
7	1	Jelene	Introduction to Business Analytics	3.6000000238
8	10	Steffie	Introduction to Business Analytics	3.6000000238
9	3	Lorelei	Data Structures and Java	3.3999999762
10	9	Coretta	Data Structures and Java	3.3999999762



```
with cte_student as (
    select s.student_id ,
           s.first_name,
           s.last_name,
           g.grade_points,
           c.course_name,
           rank () over (partition by g.grade_points order by g.grade_points desc nulls last) as order_rank
    from student as s
    inner join register as r on s.student_id = r.student_id
    inner join grade g on g.course_id = r.course_id
    inner join course c on c.course_id = g.course_id
)
select distinct cte.student_id, cte.first_name as student_first_name,
       cte.last_name as student_last_name, cte.course_name, cte.grade_points as gpa
from cte_student as cte
where cte.order_rank = 1
group by cte.grade_points, cte.course_name, cte.first_name, cte.last_name, cte.student_id;
```

student(+) 1
with cte_student as (select s.studen
Enter a SQL expression to filter results (use Ctrl+Space)

	student_id	student_first_name	student_last_name	course_name	gpa
1	1	Jelene	Fletcher	Database Theory and Design	4
2	1	Jelene	Fletcher	Introduction to Business Analytics	3.8
3	2	Kittie	Hemerijk	Database Theory and Design	4
4	3	Lorelei	Clowley	Data Structures and Java	3.5
5	4	Livvy	Clixby	Artificial Intelligence	3.8
6	5	Paulie	Chess	Software Engineering	4
7	6	Brenden	Taborre	Business Development	3.3
8	6	Brenden	Taborre	Software Engineering	4
9	7	Bertrando	Forrest	Artificial Intelligence	3.8
10	7	Bertrando	Forrest	Cyber Security	3.75
11	8	Fred	Leahy	Statistics and Predictive Analysis	3.94
12	9	Coretta	Challicombe	Data Engineering	4
13	9	Coretta	Challicombe	Data Structures and Java	3.5
14	10	Steffie	Moiser	Data Visualization	3.8
15	10	Steffie	Moiser	Introduction to Business Analytics	3.8