**Database Build Task (using MySQL)**

*Creating the Database*

CREATE DATABASE college\_data;

USE college\_data;

*Creating the Tables*

CREATE TABLE students(

-> StudentID mediumint NOT NULL UNIQUE,

-> StudentName char(100) NOT NULL,

-> ExamScore tinyint,

-> Support char(3) NOT NULL,

-> DOB date NOT NULL,

-> PRIMARY KEY (StudentID));

CREATE TABLE teachers(

-> TeacherID mediumint NOT NULL UNIQUE,

-> TeacherName char(100) NOT NULL,

-> PRIMARY KEY (TeacherID));

CREATE TABLE exam\_boards(

-> ExamBoardID mediumint NOT NULL UNIQUE,

-> ExamBoardName char(100) NOT NULL,

-> PRIMARY KEY (ExamBoardID));

CREATE TABLE courses(

-> CourseID mediumint NOT NULL UNIQUE,

-> CourseName char(100) NOT NULL,

-> TeacherID mediumint NOT NULL,

-> PRIMARY KEY (CourseID),

-> FOREIGN KEY (TeacherID) REFERENCES teachers(TeacherID));

CREATE TABLE student\_records(

-> StudentID mediumint NOT NULL,

-> CourseID mediumint NOT NULL,

-> ExamBoardID mediumint NOT NULL,

-> CONSTRAINT Student\_PK PRIMARY KEY (StudentID, CourseID, ExamBoardID),

-> FOREIGN KEY (StudentID) REFERENCES students(StudentID),

-> FOREIGN KEY (CourseID) REFERENCES courses(CourseID),

-> FOREIGN KEY (ExamBoardID) REFERENCES exam\_boards(ExamBoardID));

*Inserting Data Into Tables*

INSERT INTO students (StudentID, StudentName, ExamScore, Support, DOB)

VALUES (1002, 'Sally Davies', 55, 'yes', '1999-10-02'),

(1003, 'Mark Hanmill', 90, 'no', '1995-06-05'),

(1004, 'Anas Ali', 70, 'no', '1980-08-03'),

(1005, 'Cheuk Yin', 45, 'yes', '2002-05-01');

INSERT INTO teachers (TeacherID, TeacherName)

VALUES (1, 'Mr Jones'),

(2, 'Ms Parker'),

(3, 'Mr Peters'),

(4, 'Mrs Patel'),

(5, 'Mr Daniels');

INSERT INTO courses (CourseID, CourseName, TeacherID)

VALUES (101, 'Computer Science', 1),

(102, 'Maths', 2),

(103, 'Physics', 3),

(104, 'Biology', 4),

(105, 'Music', 5);

INSERT INTO exam\_boards (ExamBoardID, ExamBoardName)

VALUES (1, 'BCS'),

(2, 'EdExcel'),

(3, 'AQA'),

(4, 'OCR'),

(5, 'WJEC');

INSERT INTO student\_records (StudentID, CourseID, ExamBoardID)

VALUES (1001, 101, 1),

(1001, 102, 2),

(1001, 103, 4),

(1002, 102, 3),

(1002, 104, 5),

(1002, 105, 3),

(1003, 101, 1),

(1003, 102, 2),

(1003, 103, 4),

(1004, 102, 3),

(1004, 103, 4),

(1004, 104, 5),

(1005, 101, 1),

(1005, 102, 2),

(1005, 105, 3);

*Alternatively, this code also worked (exported from excel into a csv file, loaded into MySQL):*

LOAD DATA INFILE '/home/codio/workspace/stu

dent-records.csv'

INTO TABLE student\_records

FIELDS TERMINATED BY ','

LINES TERMINATED BY '\n'

IGNORE 1 LINES;

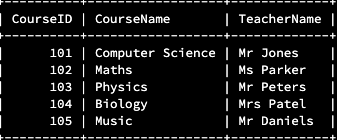
*Testing the referential integrity*

*Starting with the Teacher ID reference in the courses table – a query was successfully carried out to pull the teacher names through with the CourseID and CourseName:*

SELECT courses.CourseID, courses.CourseName, teachers.TeacherName

FROM courses

INNER JOIN teachers ON courses.TeacherID = teachers.TeacherID;



*Joining all data referenced by IDs in the student\_records table:*

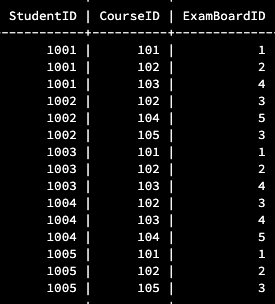
SELECT student\_records.StudentID, students.StudentName, courses.CourseName, exam\_boards.ExamBoardName

FROM student\_records

INNER JOIN students ON student\_records.StudentID = students.StudentID

INNER JOIN courses ON student\_records.CourseID = courses.CourseID

INNER JOIN exam\_boards ON student\_records.ExamBoardID = exam\_boards.ExamBoardID;



*Performing multiple joins, with reference to the student\_records table to create a view that replicates the original (un-normalized) data:*

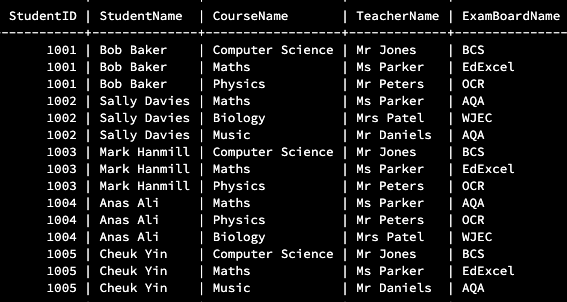
SELECT student\_records.StudentID, students.StudentName, courses.CourseName, teachers.TeacherName, exam\_boards.ExamBoardName

FROM student\_records

INNER JOIN students ON student\_records.StudentID = students.StudentID

INNER JOIN courses ON student\_records.CourseID = courses.CourseID

INNER JOIN exam\_boards ON student\_records.ExamBoardID = exam\_boards.ExamBoardID INNER JOIN teachers ON courses.TeacherID = teachers.TeacherID;

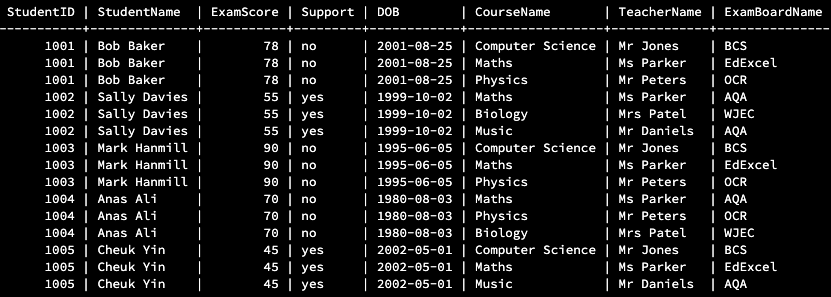


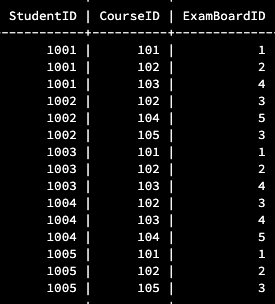
SELECT student\_records.StudentID, students.StudentName, students.ExamScore, students.Support, students.DOB, courses.CourseName, teachers.TeacherName, exam\_boards.ExamBoardName

FROM student\_records

INNER JOIN students ON student\_records.StudentID = students.StudentID

INNER JOIN courses ON student\_records.CourseID = courses.CourseID

INNER JOIN exam\_boards ON student\_records.ExamBoardID = exam\_boards.ExamBoardID INNER JOIN teachers ON courses.TeacherID = teachers.TeacherID;

**SELECT \* FROM student\_records;