Green River College

Final Project

School Scores Database

Blezyl Santos

IT 201 Database Fundamentals

December 11, 2019

Analysis

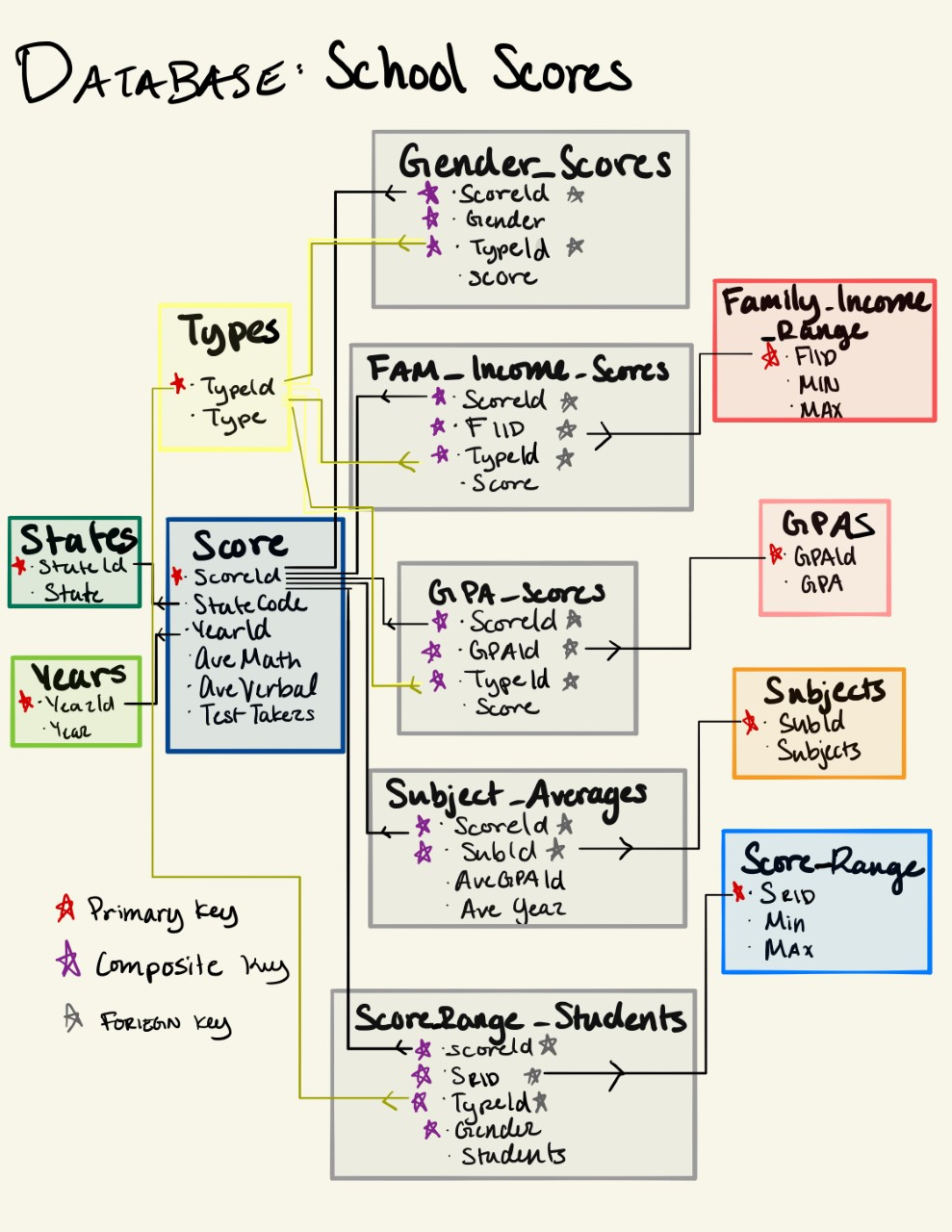
The school score database shows the amount of test takers and average math and verbal scores for the SAT test. The database consists of scores from all states in the USA for each year from 2005 to 2015. There are 7 look-up table containing a name or description and a corresponding id. The look-up tables are *family-income-range*, *gpas*, *score\_range,* *types, subjects, state*, and *year*. There are also 6 linking tables that consists of foreign keys from the look-up tables and contains the results (i.e. verbal and math score, and test takers). The linking tables are *Fam\_income\_scores, gpa\_scores, score\_range\_students, gender\_scores, subject\_averages,* and *score*. The gpa\_scores table is the only table that does not have average math and verbal scores and test takers, but it does have average gpa and average year.

As a student, it is interesting to see how other students have done on the same test. Also, this dataset includes other factors other than GPAs, it includes family income, gender, and all of the states spanning 10 years.

Questions

1. Display the scores of states that contain the word “New” and has an Average Math score between 550-600
2. Display the top 5 Average Verbal scores and years of Texas with a GPA of “A+”, “A-”, or “A.”
3. Show the States and average female math scores that are more than the average score of males in 2015.
4. What Family income range has the most test takers in California on 2010.
5. Show the state, average gpa, average verbal score for the highest average GPA in the subject of “Arts/Music” for each year.
6. Show the Total number of male students in each score range on 2013 that is greater than 5000.

Design



Implementation

Table for original database

CREATE TABLE schoolscores (

id int(4) NULL NULL PRIMARY KEY AUTO\_INCREMENT,

year year,

staeCode char(2),

state varchar(30),

AveMath int(3),

AveVerbal int(3),

TestTakers int(7),

AcaArtGPA decimal(3,2),AcaArtYear decimal(3,2),

AcaEngGPA decimal(3,2),AcaEngYear decimal(3,2),

AcaForeignGPA decimal(3,2), AcaForeignYear decimal(3,2),

AcaMathGPA decimal(3,2),AcaMathYear decimal(3,2),

AcaNatGPA decimal(3,2)AcaNatYear decimal(3,2),

AcaSocGPA decimal(3,2),AcaSocYear decimal(3,2),

fi2M int(3),fi2t int(5),fi2v int(3),

fi4M int(3),fi4t int(5),fi4v int(3),

fi6M int(3),fi6t int(5),fi6v int(3),

fi8M int(3),fi8t int(5),fi8v int(3),

fi0M int(3),fi0t int(5),fi0v int(3),

fi1M int(3),fi1t int(5),fi1v int(3),

gpa\_amM int(3),gpa\_amT int(5),gpa\_amV int(3),

gpa\_apM int(3),gpa\_apT int(5),gpa\_apV int(3),

gpa\_aM int(3),gpa\_aT int(5),gpa\_aV int(3),

gpa\_bM int(3),gpa\_bT int(5),gpa\_bV int(3),

gpa\_cM int(3),gpa\_cT int(5),gpa\_cV int(3),

gpa\_dM int(3),gpa\_dT int(5),gpa\_dV int(3),

gpa\_nM int(3),gpa\_nT int(5),gpa\_nV int(3),

genderFM int(3),genderFT int(5),genderFV int(3),

genderMM int(3),genderMT int(5),genderMV int(3),

scr2MF int(4),scr2MM int(4),scr2MT int(4),

scr2vF int(4),scr2vM int(4),scr2vT int(4),

scr3MF int(4),scr3MM int(4),scr3MT int(4),

scr3vF int(4),scr3vM int(4),scr3vT int(4),

scr4MF int(4),scr4MM int(4),scr4MT int(4),

scr4vF int(4),scr4vM int(4),scr4vT int(4),

scr5MF int(4),scr5MM int(4),scr5MT int(4),

scr5vF int(4),scr5vM int(4),scr5vT int(4),

scr6MF int(4),scr6MM int(4),scr6MT int(4),

scr6vF int(4),scr6vM int(4),scr6vT int(4),

scr7MF int(4),scr7MM int(4),scr7MT int(4),

scr7vF int(4),scr7vM int(4),scr7vT int(4)

)

7 look-up tables

--------------------------------------------------------------------------

CREATE TABLE states (

stateCode char(2) NOT NULL PRIMARY KEY,

state varchar(30)

)

INSERT INTO states (states.stateCode, states.state) SELECT DISTINCT schoolscores.staeCode, schoolscores.state FROM schoolscores

--------------------------------------------------------------------------

CREATE TABLE years(

yearId int(2) NOT NULL PRIMARY KEY AUTO\_INCREMENT,

year year

)

INSERT INTO years (year) SELECT DISTINCT schoolscores.year FROM schoolscores

--------------------------------------------------------------------------

CREATE TABLE types(

typeId int(2) NOT NULL PRIMARY KEY AUTO\_INCREMENT,

type VARCHAR(10)

)

INSERT INTO types(type) VALUES ("Math"), ("Verbal"), ("Test-Takers")

--------------------------------------------------------------------------CREATE TABLE family\_income\_range(

fiid int(1) NOT NULL PRIMARY KEY AUTO\_INCREMENT,

min int(3),

max int(3)

)

INSERT INTO family\_income\_range (min, max) VALUES (0,20),(20,40),(40,60),(60,80), (80,100), (100,999)

--------------------------------------------------------------------------

CREATE TABLE gpas(

gpaid int(1) NOT NULL PRIMARY KEY AUTO\_INCREMENT,

gpa char(15)

)

INSERT INTO gpas(gpa) VALUES ("A+"), ("A"), ("A-"),("B"),("C"),("D or Lower"), ("No response")

--------------------------------------------------------------------------

CREATE TABLE score\_range(

srid int(1) NOT NULL PRIMARY KEY AUTO\_INCREMENT,

min int(3),

max int(3)

)

INSERT INTO score\_range (min, max) VALUES (200,300),(300,400),(400,500),(500,600),(600,700), (700,800)

--------------------------------------------------------------------------

CREATE TABLE subjects(

subid int(1) NOT NULL PRIMARY KEY AUTO\_INCREMENT,

subject varchar(30)

)

INSERT INTO subjects (subject) VALUES

("Arts/Music"),("English"),("Foreign Language"),("Mathematics"),("Natural Science"),("Social Science")

--------------------------------------------------------------------------

6 Linking tables

Only included one insert-update statement to reduce pages. To insert more data from the original table, change the where clause and the column name from schoolscores

(e.g.

INSERT ...

where fiid = 1 AND typeid = 1;

UPDATE ...

SET Fam\_income\_scores.score = schoolscores.fi0M

where fiid = 1 AND typeid = 1 ;

to

INSERT ...

where fiid = *4* AND typeid = *3*;

...

UPDATE ...

SET Fam\_income\_scores.score = schoolscores.*fi6T*

where fiid = *4* AND typeid = *3* ;

)

--------------------------------------------------------------------------

CREATE TABLE Fam\_income\_scores(

scoreId int(3),

fiid int(1),

typeId int(2),

score int(7),

FOREIGN KEY (scoreId) REFERENCES score(scoreId),

FOREIGN KEY (typeId) REFERENCES types(typeId),

FOREIGN KEY (fiid) REFERENCES family\_income\_range(fiid),

PRIMARY KEY (scoreId, fiid, typeid)

)

INSERT INTO Fam\_income\_scores (scoreid, fiid, typeId)

SELECT DISTINCT score.scoreId, family\_income\_range.fiid, types.typeId

FROM score, family\_income\_range, types

where fiid = 1 AND typeid = 1

ORDER BY `score`.`scoreId` ASC;

UPDATE Fam\_income\_scores

INNER JOIN score ON

score.scoreId=Fam\_income\_scores.scoreId

INNER JOIN years ON

years.yearId = score.yearsId

INNER JOIN schoolscores ON

score.stateCode = schoolscores.staeCode AND

years.year = schoolscores.year

SET Fam\_income\_scores.score = schoolscores.fi0M

where fiid = 1 AND typeid = 1;

...

--------------------------------------------------------------------------

CREATE TABLE gpa\_scores(

scoreId int(3),

gpaid int(1),

typeId int(2),

score int(7),

FOREIGN KEY (scoreId) REFERENCES score(scoreId),

FOREIGN KEY (gpaId) REFERENCES gpas(gpaId),

FOREIGN KEY (typeid) REFERENCES types(typeId),

PRIMARY KEY (scoreId, gpaid, typeid)

)

INSERT gpa\_scores(scoreId, gpaid, typeid)

SELECT DISTINCT score.scoreId, gpas.gpaid, types.typeId

FROM score, gpas, types

WHERE gpaid = 1 and typeid = 1;

UPDATE gpa\_scores

INNER JOIN score ON

score.scoreId=gpa\_scores.scoreId

INNER JOIN years ON

years.yearId = score.yearsId

INNER JOIN schoolscores ON

score.stateCode = schoolscores.staeCode AND

years.year = schoolscores.year

SET gpa\_scores.score = schoolscores.gpa\_apM

WHERE gpaid = 1 and typeid = 1;

...

--------------------------------------------------------------------------

CREATE TABLE score\_range\_students(

scoreId int(3),

srid int(1),

typeid int(2),

gender varchar(10),

students int(5),

FOREIGN KEY (scoreId) REFERENCES score(scoreId),

FOREIGN KEY (srid) REFERENCES score\_range(srid),

FOREIGN KEY (typeid) REFERENCES types(typeid),

PRIMARY key (scoreid, srid, typeid, gender)

)

​INSERT INTO score\_range\_students(scoreId, srid, typeid, gender)

SELECT DISTINCT score.scoreId, score\_range.srid, types.typeId, gender\_scores.gender

FROM score, score\_range, types, gender\_scores

WHERE score\_range.srid = 1 and types.typeId = 1 and gender\_scores.gender = "Female"

ORDER BY `score`.`scoreId` ASC;

UPDATE score\_range\_students

INNER JOIN score ON

score.scoreId=score\_range\_students.scoreId

INNER JOIN years ON

years.yearId = score.yearsId

INNER JOIN schoolscores ON

score.stateCode = schoolscores.staeCode AND

years.year = schoolscores.year

SET score\_range\_students.students = schoolscores.scr2MF

WHERE srid = 1 and typeId = 1 and gender = "Female";

...

--------------------------------------------------------------------------

CREATE TABLE gender\_scores(

scoreId int(3),

gender varchar(6),

typeId int(2),

score int(5),

PRIMARY KEY (scoreId,typeId,gender),

FOREIGN KEY (scoreId) REFERENCES score(scoreId),

FOREIGN KEY (typeId) REFERENCES types(typeId)

)

INSERT INTO gender\_scores(scoreId, typeid, gender)

SELECT DISTINCT score.scoreId, types.typeId, score\_range\_students.gender

FROM score, types, score\_range\_students

WHERE types.typeId = 1 and score\_range\_students.gender = "Male"

ORDER BY `score`.`scoreId` ASC;

UPDATE gender\_scores

INNER JOIN score ON

score.scoreId=gender\_scores.scoreId

INNER JOIN years ON

years.yearId = score.yearsId

INNER JOIN schoolscores ON

score.stateCode = schoolscores.staeCode AND

years.year = schoolscores.year

SET gender\_scores.score = schoolscores.genderMM

WHERE typeId = 1 and gender = "Male";

...

--------------------------------------------------------------------------

CREATE TABLE subject\_averages(

scoreId int(3),

subid int(1),

aveGPA decimal(4,2),

aveYear decimal(4,2),

FOREIGN KEY (scoreId) REFERENCES score(scoreId),

FOREIGN KEY (subId) REFERENCES subjects(subId),

PRIMARY KEY (scoreId, subid)

)

INSERT INTO subject\_averages(scoreId, subid)

SELECT DISTINCT score.scoreId, subjects.subid

FROM score, subjects

WHERE subjects.subid = 1;

UPDATE subject\_averages

INNER JOIN schoolscores ON

subject\_averages.aveGPA=schoolscores.AcaArtGPA AND

subject\_averages.aveYear=schoolscores.AcaArtYear

INNER JOIN score ON

score.stateCode = schoolscores.staeCode

INNER JOIN years ON

years.yearId = score.yearsId AND

years.year = schoolscores.year

SET subject\_averages.scoreId = score.scoreId

WHERE subid=1;

...

--------------------------------------------------------------------------

CREATE TABLE score(

scoreId int(3) NOT NULL PRIMARY KEY AUTO\_INCREMENT,

stateCode char(2),

yearsId int(2),

AveMath int(3),

AveVerbal int(3),

TestTakers int(7),

FOREIGN KEY (stateCode) REFERENCES states(stateCode),

FOREIGN KEY (yearsId) REFERENCES years(yearId)

)

INSERT INTO score(AveMath, AveVerbal, TestTakers)

SELECT schoolscores.AveMath, schoolscores.AveVerbal, schoolscores.TestTakers

FROM schoolscores

================================================================

UPDATE score

INNER JOIN schoolscores ON score.AveMath=schoolscores.AveMath AND score.AveVerbal = schoolscores.AveVerbal AND schoolscores.TestTakers = score.TestTakers

INNER JOIN states ON states.stateCode=schoolscores.staeCode AND states.state = schoolscores.state

SET score.stateCode = states.stateCode

UPDATE score

INNER JOIN schoolscores ON score.AveMath=schoolscores.AveMath AND

score.AveVerbal = schoolscores.AveVerbal AND

schoolscores.TestTakers = score.TestTakers

INNER JOIN years ON years.year=schoolscores.year

SET score.yearsId = years.yearId

--------------------------------------------------------------------------

Utilization

1. Display the years, states, and Average Math scores of states that contain the word “New” and has an Average Math score between 525-550. And ordered from least to greatest.

SELECT states.state, score.AveMath, years.year

FROM score, states, years

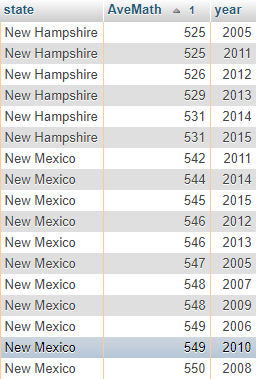
WHERE states.stateCode = score.stateCode

AND score.yearsId = years.yearId

AND state LIKE "%New%"

AND AveMath BETWEEN 525 AND 550

ORDER BY `score`.`AveMath` ASC



1. Display the top 5 Average Verbal scores, year, and GPA of Texas with a GPA of “A+”, “A-”, or “A.”

SELECT years.year, gpa\_scores.score, gpas.gpa

FROM years, score, gpa\_scores, states, gpas, types

WHERE years.yearId = score.yearsId

AND gpa\_scores.scoreId = score.scoreId

AND states.stateCode = score.stateCode

AND gpas.gpaid = gpa\_scores.gpaid

AND types.typeId = gpa\_scores.typeId

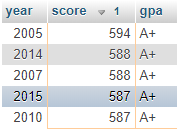
AND state = "Texas"

AND gpas.gpa IN ("A+","A","A-")

AND type = "Verbal"

ORDER BY `gpa\_scores`.`score` DESC

LIMIT 5



1. Show the States and average female math scores that are more than the average math score of males in all states on 2010.

SELECT state, gender\_scores.score, gender\_scores.typeId

FROM `gender\_scores`, years, score, types, states

WHERE gender\_scores.scoreId = score.scoreId

AND years.yearId = score.yearsId

AND types.typeId = gender\_scores.typeId

AND states.stateCode = score.stateCode

AND year = "2010"

AND gender = "female"

AND type = "Math"

AND score > (

SELECT AVG(score)

FROM `gender\_scores`, years, score, types, states

WHERE gender\_scores.scoreId = score.scoreId

AND years.yearId = score.yearsId

AND types.typeId = gender\_scores.typeId

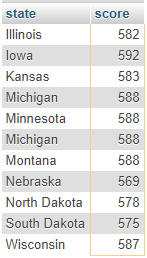
AND states.stateCode = score.stateCode

AND gender = "male"

AND types.type = "Math"

AND year = "2010"

)



1. What Family income range has the most test takers in California on 2010.

SELECT family\_income\_range.min, family\_income\_range.max, score

FROM `Fam\_income\_scores`, family\_income\_range

WHERE Fam\_income\_scores.fiid = family\_income\_range.fiid

AND score = (

SELECT MAX(Fam\_income\_scores.score)

FROM `Fam\_income\_scores`, years, score, types, states

WHERE Fam\_income\_scores.scoreId = score.scoreId

AND years.yearId = score.yearsId

AND types.typeId = Fam\_income\_scores.typeId

AND states.stateCode = score.stateCode

AND state = "california"

AND year = "2010"

)



1. Show the state, average gpa, average verbal score for the highest average GPA in the subject of “Natual Science” for each year.

SELECT state, year, aveGPA, score.AveVerbal

FROM subject\_averages, years, score, subjects, states

WHERE subject\_averages.scoreId = score.scoreId

AND years.yearId = score.yearsId

AND states.stateCode = score.stateCode

AND subjects.subid = subject\_averages.subid

AND (year, aveGPA, subject) IN (

SELECT year, MAX(aveGPA), subjects.subject

FROM subject\_averages, years, score, subjects

WHERE subject\_averages.scoreId = score.scoreId

AND score.yearsId = years.yearId

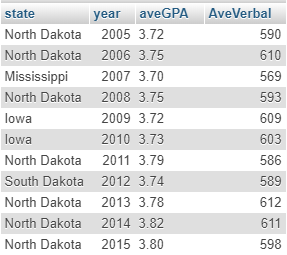
AND subject\_averages.subid = subjects.subid

AND subject = "Natural Science"

GROUP BY year

)

GROUP BY year



1. Show the Total number of male students that took the math test in each score range on 2013 that is greater than 100,000.

SELECT score\_range.min as 'Score Range From', score\_range.max as 'To', SUM(students)

FROM `score\_range\_students`, types, states, score, years, score\_range

WHERE types.typeId =score\_range\_students.typeid

AND types.typeId = score.yearsId

AND states.stateCode = score.stateCode

AND score\_range\_students.scoreId = score.scoreId

AND score\_range.srid = score\_range\_students.srid

AND gender = "male"

AND types.type = "Math"

AND year = 2013

GROUP BY score\_range.srid

HAVING sum(students) > 100000

