Acme Database System

Zach Miles

Database Management Concepts

COSC – 4385.001

Dr. Brown

**TABLE OF CONTENTS**

Description PAGE

E-R Diagram 4

Data Dictionary 5

SQL Statements for Creating Tables 9

SQL Statements for Dropping Tables 21

Relational Algebra Statements 23

Query 7 23

Query 9 23

Query 13 23

Query 14 23

Query 15 23

Query 16 23

Query 17 24

Query 18 24

SQL Statements 25

Query 1 25

Query 2 26

Query 3 29

Query 4 33

Query 5 37

Query 6 38

Query 7 40

Query 8 40

Query 9 41

Query 10 42

Query 11 42

Query 12 44

Query 13 44

Query 14 44

Query 15 44

Query 16 45

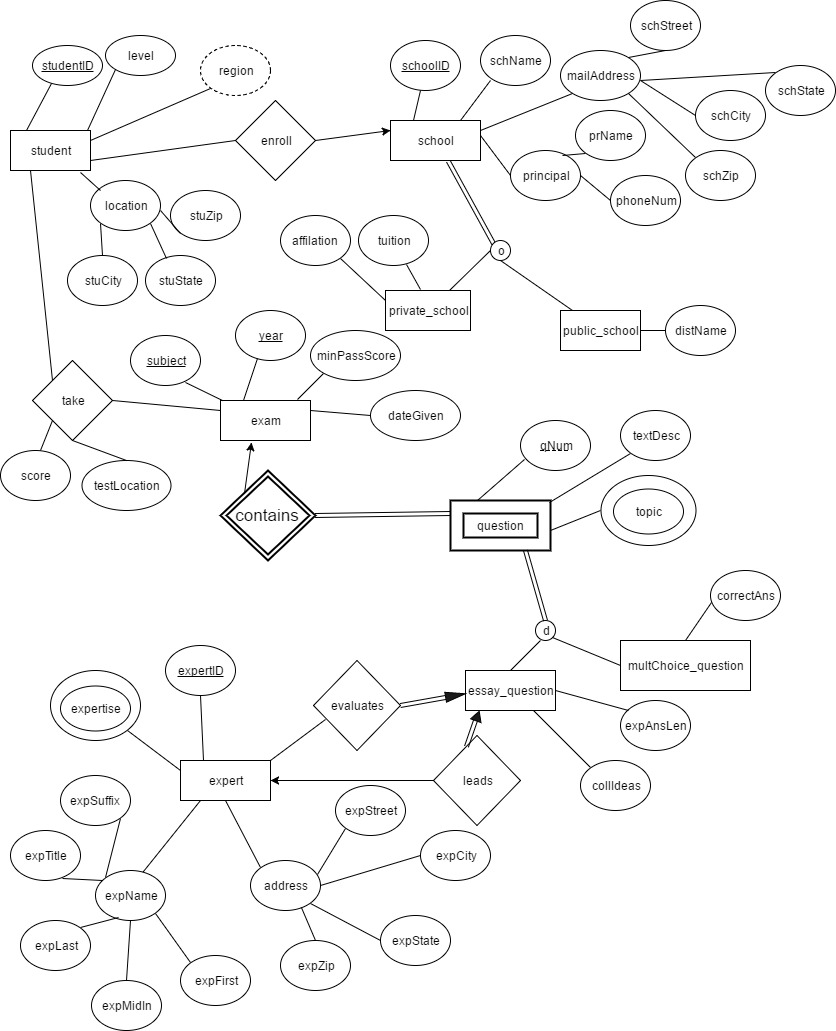
Query 17 45

Query 18 45

Query 19 46

Query 20 46

**E-R Diagram**



**Data Dictionary**

**Table:** student (studentID, level, stuCity, stuState, stuZip)

**Attributes**

studentID – number

stuLevel – varchar2(15)

stuCity – varchar2(15)

stuState – varchar2(12)

stuZip – number

constraint: stuLevel must be Freshman, Sophomore, Junior, or Senior

constraint: studentID and stuZip must be greater than 0

**Table:** school (schoolID, schName, principalName, principalPNum, schStreet, schAddress, schCity, schZip)

**Attributes**

schoolID – number

schName - varchar2(25)

principalName - varchar2(20)

principalPNum – vachar2(15)

schStreet - varchar2(12)

schCity - varchar2(20)

schState – varchar2(20)

schZip – number

constraint: schZip must be greater than 0

**Table:** private\_school (schoolID, tuition, affiliation)

**Attributes**

schoolID – number

tuition – number(10,2)

affiliation – varchar2(12)

constraint: tuition must be greater than or equal to 0

constraint: schoolID must be greater than 0

foreign key schoolID references school

**Table:** public\_school (schoolID, distName)

**Attributes**

schoolID – number

distName – varchar2(18)

foreign key schoolID references school

**Table:** enroll (schoolID, studentID)

**Attributes**

schoolID – number

studentID – number

foreign key studentID references student

**Table:** exam (subject, year, minPassScore, dateGiven)

**Attributes**

subject – varchar2(12)

year – number

minPassScore – number

dateGiven – date

constraint: year must be greater than or equal to 2000

constraint: minPassScore must be greater than or equal to 0

**Table:** takes (studentID, year, subject, score, testLocation)

**Attributes**

studentID - number

subject – varchar2(12)

year – number

score – number

testLocation – varchar2(25)

constraint: score must be greater than or equal to 0

foreign key studentID references student

foreign key (subject, year) references exam

**Table:** essay\_question (year, subject, qNum, collIdeas, expAnsLen)

**Attributes**

subject – varchar2(12)

year – number

qNum – number

collIdeas – varchar2(30)

expAnsLen – number,

constraint: qNum must be greater than or equal to one

constraint: year must be greater than or equal to 2000

**Table:** multChoice\_question (year, subject, qNum, correctAns)

**Attributes**

subject – varchar2(12)

year – number

qNum – number

correctAns – char(1)

constraint: qNum must be greater than or equal to one

constraint: year must be greater than or equal to 2000

**Table:** question\_topic (year, subject, qNum, topic)

**Attributes**

subject – varchar2(12)

year – number

qNum – number

topic – varchar2(25),

constraint: qNum must be greater than or equal to one

**Table:** expert (expertID, expSuffix, expTitle, expLast, expMidln, expFirst, expStreet, expCity, expState, expZip)

**Attribute**

expertID number

expSuffix – varchar2(5)

expTitle ­– varchar2(10)

expLast – varchar2(15)

expMidln – varchar(1)

expFirst – varchar2(15)

expStreet – varchar2(15)

expCity – varchar2(15)

expState – varchar2 (12)

expZip – number

constraint: expZip must be greater than 0

constraint: expertID must be greater than 0

**Table:** expert\_evaluates (expertID, year, subject, qNum)

**Attributes**

expertID – number

year – number

qNum – number

subject – varchar2(20)

constraint: qNum must be greater than or equal to 1

foreign key expertID references expert

foreign key (year, subject, qNum) references essay\_question

**Table:** expert\_leads (expertID, year, subject, qNum)

**Attributes**

expertID – number

year – number

qNum – number

subject – varchar2(20)

constraint: qNum must be greater than or equal to 1

foreign key expertID references expert

foreign key (year, subject, qNum) references essay\_question

**Table:** expert\_expertise (expertID, expertise)

**Attributes**

expertID - number

expertise – varchar2(15)

foreign key expertID references expert

**SQL Statements for Creating Tables**

create table student(

studentID number,

stuLevel varchar2(15),

stuCity varchar2(15),

stuState varchar2(12),

stuZip number,

primary key (studentID),

check (stuLevel IN ('Freshman', 'Sophomore', 'Junior', 'Senior')),

check (studentID > 0),

check (stuZip > 0)

);

create table school(

schoolID number,

schName varchar2(25),

principalName varchar2(20),

principalNum varchar2(15),

schStreet varchar2(12),

schCity varchar2(20),

schState varchar2(20),

schZip number,

primary key (schoolID),

check (schZip > 0)

);

create table private\_school(

schoolID number,

tuition number(10,2),

affiliation varchar2(12),

primary key (schoolID),

foreign key(schoolID) references school on delete cascade,

check (tuition >= 0),

check (schoolID > 0)

);

create table public\_school(

schoolID number,

distName varchar2(18),

primary key (schoolID),

foreign key(schoolID) references school on delete cascade

);

create table enroll(

schoolID number,

studentID number,

primary key (studentID),

foreign key(studentID) references student on delete cascade

);

create table exam(

subject varchar2(12),

year number,

minPassScore number,

dateGiven date,

primary key (subject, year),

check (minPassScore >= 0),

check (year >= 2000)

);

create table takes(

studentID number,

subject varchar2(12),

year number,

score number,

testLocation varchar2(25),

primary key (studentID, subject, year),

foreign key(studentID) references student on delete cascade,

foreign key(subject, year) references exam on delete cascade,

check (score >= 0)

);

create table essay\_question(

year number,

subject varchar2(12),

qNum number,

collIdeas varchar2(30),

expAnsLenLines number,

primary key (year, subject, qNum),

check (year >= 2000),

check (qNum >= 1)

);

create table multChoice\_question(

year number,

subject varchar2(12),

qNum number,

correctAns varchar2(1),

primary key (year, subject, qNum),

check (year >= 2000),

check (qNum >= 1)

);

create table question\_topic(

year number,

subject varchar2(12),

qNum number,

topic varchar2(25),

primary key (year, subject, qNum, topic),

check (qNum >= 1),

foreign key (year, subject, qNum) references essay\_question on delete cascade,

foreign key (year, subject, qNum) references multChoice\_question on delete cascade

);

create table expert(

expertID number,

expSuffix varchar2(5),

expTitle varchar2(10),

expLast varchar2(15),

expMidIn varchar2(1),

expFirst varchar2(15),

expStreet varchar2(15),

expCity varchar2(15),

expState varchar2(12),

expZip number,

primary key (expertID),

check (expZip > 0)

);

create table expert\_evaluates(

expertID number,

year number,

qNum number,

subject varchar2(20),

primary key (expertID),

foreign key (year, subject, qNum) references essay\_question on delete cascade,

foreign key (expertID) references expert on delete cascade

);

create table expert\_leads(

expertID number,

year number,

qNum number,

subject varchar2(20),

primary key (expertID),

foreign key (year, subject, qNum) references essay\_question on delete cascade,

foreign key (expertID) references expert on delete cascade

);

create table expert\_expertise(

expertID number,

expertise varchar2(15),

primary key (expertID, expertise),

foreign key(expertID) references expert on delete cascade

);

create table region(

state varchar2(20),

region varchar2(20),

primary key (state)

);

create

view

StudentMathScores

as

select year, studentID, schoolID, score, subject, minPassScore

from

(select year, studentID, schoolID, score, subject,

minPassScore

from enroll natural join takes natural join exam

where subject = 'Math' AND year = 2017)

where schoolID in (select schoolID from public\_school);

create

view

passExam

as

select count(studentID) numPass, schoolID

from studentmathscores

where (score > minpassscore)

group by schoolID;

create

view

totalExamAttempt

as

select count(studentID) numAttempt, schoolID

from studentmathscores

group by schoolID;

create

view

EssayQTopics

as

(select topic, count(qNum) NumEssayQs

from

(Select \*

from essay\_question natural join question\_topic

where year = (select to\_number(to\_char(sysdate, 'YYYY')) from dual))

group by topic);

create

view

MultChQTopics

as

(select topic, count(qNum) NumMultCQs

from

(select \*

from multChoice\_question natural join question\_topic

where year = (select to\_number(to\_char(sysdate, 'YYYY')) from dual))

group by topic);

create view

PrivateSchoolTestScores

as

select \* from

(select \* from takes where studentID in (select studentID

from private\_school))

natural join exam

natural join enroll

where year = 2017

and schoolID in (select schoolID from private\_school)

and schoolID not in (select schoolID from public\_school);

create view

regionMap

as

select studentID, stuState, region

from

(student natural natural join region)

where stuState = state;

create

view

homeschooled

as

select studentID

from (select studentID, stuState

from student

where studentID not in (select studentID

from student natural join enroll));

create

view

turbo

as

select studentID, stuState, region,

case when studentID in (select StudentID from homeschooled) then 'yes'

else 'no' end as homeschooled

from regionmap

order by region;

create

view

essayQexam

as

select year, subject, minPassScore, studentId, score,

case when score >=minpassscore then 'yes'

else 'no' end as passed

from essay\_question natural join exam natural join takes natural join student;

create

view

EssayExamPassPercent

as

select x.year, x.subject, (pass/total) percentPass

from

(select year, subject, count(\*) total

from essayqexam

group by year, subject)x,

(select year, subject, count(\*) pass

from essayqexam

where score >= minpassscore

group by year, subject)y

where x.subject = y.subject AND

x.year = y.year;

create

view

publicmathscores2016

as

select studentID, score, schoolID, schName, minPassScore

from takes

natural join student

natural join enroll

natural join school

natural join exam

where subject = 'Math'

and year = 2016

and schoolID in (select schoolID from publicschools);

create

view

publicmathscores2017

as

select studentID, score, schoolID, schName, minPassScore

from takes

natural join student

natural join enroll

natural join school

natural join exam

where subject = 'Math'

and year = 2017

and schoolID in (select schoolID from publicschools);

create

view

percentchange

as

select schoolID, ((percent17-percent16)\*100) dif

from

(select schoolID, (numPass17/total17) percent17

from

(select schoolID, count(\*) numPass17

from

publicmathscores2017

where score >= minpassscore

group by schoolID)

natural join

(select schoolID, count(\*) total17

from publicmathscores2017

group by schoolID))

natural join

(select schoolID, (numPass16/total16) percent16

from

(select schoolID, count(\*) numPass16

from

publicmathscores2016

where score >= minpassscore

group by schoolID)

natural join

(select schoolID, count(\*) total16

from publicmathscores2016

group by schoolID));

create

view

StateSciencePassRate

as

(select y.schoolID, y.schstate, (numpass/numattempt \*100) percentpass

from

(select schoolID, schstate, count(schoolID) numPass

from

(

select subject, year, schoolID, studentID, schstate, score, minpassscore

from takes natural join exam natural join school

natural join enroll

where subject = 'Science')

where score >= minpassscore

group by schoolID, schstate)x,

(select schoolID, schstate, count(schoolID) numAttempt

from

(

select subject, year, schoolID, studentID, schstate, score, minpassscore

from takes natural join exam natural join school

natural join enroll

where subject = 'Science')

group by schoolID, schstate)y

where x.schoolID = y.schoolID and x.schstate = y.schstate);

**SQL Statements for Dropping Tables**

drop view StateSciencePassRate;

drop view Studentmathscores;

drop view publicmathscores2016;

drop view publicmathscores2017;

drop view percentchange;

drop view passExam;

drop view totalExamAttempt;

drop view essayqtopics;

drop view multchqtopics;

drop view PrivateSchoolTestScores;

drop view regionmap;

drop view homeschooled;

drop view turbo;

drop view essayqexam;

drop view essayexampasspercent;

drop table region;

drop table private\_school;

drop table public\_school;

drop table enroll;

drop table takes;

drop table question\_topic;

drop table expert\_evaluates;

drop table expert\_expertise;

drop table expert\_leads;

drop table essay\_question;

drop table expert;

drop table multChoice\_question;

drop table student;

drop table school;

drop table exam;

**Relational Algebra Statements**

**Query 7**

∏subject,studentID,score,stuLevel,testlocation((σstuState=”Texas”,year=”2017” (student ⨝ takes)

**Query 9**

PSTS 🡨 σstudentID (private\_school)

HSS 🡨 σstudentID (public\_school)

PSTS 🡨 σyear=”2017”,studentID=PSS.StudentID,studentID≠HSS.studentID (PSS ⨝ exam ⨝ enroll)

passed 🡨 schoolIDGcount(\*)(σscore >=minPassScore (PSTS))

total 🡨 schoolIDGcount(\*) (PSTS)

ratio 🡨 ∏total.schoolID,(passed.count(\*)/total.count(\*))(σtotal.schoolID=passed.schoolID (passed x total))

max 🡨 Gmax(passed.count(\*)/total.count(\*))(ratio)

target 🡨 σ ratio.(passed.count(\*)/total.count(\*))=max(passed.count(\*)/total.count(\*))(max x ratio)

∏schName,tuition,principalName,principalPNum(private\_school natural join target)

**Query 13**

∏expTitle,expLast(σexpertise=”World History” (expert ⨝ expert\_expertise))

**Query 14**

t1 🡨 ∏subject,year,testLocation(exam ⨝ takes)

testLocationGcount(testLocation)(t1)

**Query 15**

t1 🡨 topicGcount(qnum) (σyear=”2017” (essay\_question ⨝ question\_topic))

t2 🡨 topicGcount(qnum) (σyear=”2017” (multChoice\_question ⨝ question\_topic))

∏topic,t2.count(qNum),t1.count(qNum) (t1 ⨝ t2)

**Query 16**

∏subject,year,score(σstuZip=expZip (student ⨝ takes ⨝ exam ⨝ expert\_leads ⨝ expert)

**Query 17**

exam 🡨 ∏subject,year,minPassScore+5,dateGiven(σsubject=”Science” (exam)) ∪ (exam –σsubject=”Science” (exam))

**Query 18**

t1 🡨 σschoolID(private\_school) – (σschoolID(public\_school) ∩ σschoolID(private\_school))

t2 🡨 ∏ schoolID,tuition\*1.10,affiliation(t1 ⨝ private\_school)

private\_school 🡨 t2∪ (private\_school –(t2 ∩ private\_school))

**SQL Statements**

**Query 1**

insert into student (studentID, stuLevel, stuCity, stuState, stuZip)

values (1001, 'Freshman', 'Tyler', 'Texas', 75701);

insert into student (studentID, stuLevel, stuCity, stuState, stuZip)

values (1002, 'Freshman', 'Tyler', 'Texas', 75701);

insert into student (studentID, stuLevel, stuCity, stuState, stuZip)

values (1021, 'Freshman', 'Tyler', 'Texas', 75701);

insert into student (studentID, stuLevel, stuCity, stuState, stuZip)

values (1003, 'Freshman', 'Longview', 'Texas', 75604);

insert into student (studentID, stuLevel, stuCity, stuState, stuZip)

values (1004, 'Sophomore', 'Longview', 'Texas', 75604);

insert into student (studentID, stuLevel, stuCity, stuState, stuZip)

values (1022, 'Sophomore', 'Longview', 'Texas', 75604);

insert into student (studentID, stuLevel, stuCity, stuState, stuZip)

values (1005, 'Sophomore', 'Miami', 'Florida', 33101);

insert into student (studentID, stuLevel, stuCity, stuState, stuZip)

values (1006, 'Sophomore', 'Miami', 'Florida', 33101);

insert into student (studentID, stuLevel, stuCity, stuState, stuZip)

values (1007, 'Junior', 'Miami', 'Florida', 33101);

insert into student (studentID, stuLevel, stuCity, stuState, stuZip)

values (1008, 'Junior', 'Malboro', 'Maryland', 20772);

insert into student (studentID, stuLevel, stuCity, stuState, stuZip)

values (1009, 'Junior', 'Malboro', 'Maryland', 20772);

insert into student (studentID, stuLevel, stuCity, stuState, stuZip)

values (1010, 'Senior', 'Malboro', 'Maryland', 20772);

insert into student (studentID, stuLevel, stuCity, stuState, stuZip)

values (1011, 'Senior', 'Uncertain', 'Texas', 75661);

insert into student (studentID, stuLevel, stuCity, stuState, stuZip)

values (1012, 'Junior', 'Uncertain', 'Texas', 75661);

insert into student (studentID, stuLevel, stuCity, stuState, stuZip)

values (1013, 'Freshman', 'Portland', 'Oregon', 97035);

insert into student (studentID, stuLevel, stuCity, stuState, stuZip)

values (1014, 'Freshman', 'Metropolis', 'Delaware', 22221);

insert into student (studentID, stuLevel, stuCity, stuState, stuZip)

values (1015, 'Junior', 'Metropolis', 'Delaware', 22221);

insert into student (studentID, stuLevel, stuCity, stuState, stuZip)

values (1016, 'Sophomore', 'Gotham', 'New Jersey', 98765);

insert into student (studentID, stuLevel, stuCity, stuState, stuZip)

values (1017, 'Freshman', 'Gotham', 'New Jersey', 98765);

insert into student (studentID, stuLevel, stuCity, stuState, stuZip)

values (1018, 'Freshman', 'Springfield', 'Missouri', 65800);

insert into student (studentID, stuLevel, stuCity, stuState, stuZip)

values (1019, 'Junior', 'Springfield', 'Missouri', 65800);

insert into student (studentID, stuLevel, stuCity, stuState, stuZip)

values (1020, 'Sophomore', 'Springfield', 'Missouri', 65800);

insert into student (studentID, stuLevel, stuCity, stuState, stuZip)

values (5557, 'Freshman', 'Frisco', 'Texas', 75033);

insert into student (studentID, stuLevel, stuCity, stuState, stuZip)

values (5558, 'Junior', 'Frisco', 'Texas', 75033);

insert into student (studentID, stuLevel, stuCity, stuState, stuZip)

values (5559, 'Sophomore', 'Frisco', 'Texas', 75033);

commit;

**Query 2**

insert into school(schoolID, schName, principalName, principalNum, schStreet, schCity, schState, schZip)

values (9000, 'Grace Creek', 'Mr. Darcy', '(903)123-4567', 'Easy', 'Tyler', 'Texas', 75701);

insert into enroll(schoolID, studentID)

values (9000, 1001);

insert into enroll(schoolID, studentID)

values (9000, 1002);

insert into enroll(schoolID, studentID)

values (9000, 1021);

insert into private\_school(schoolID, tuition, affiliation)

values (9000, 10000, 'Catholic');

insert into public\_school(schoolID, distName)

values (9000, 'Pratter');

insert into school(schoolID, schName, principalName, principalNum, schStreet, schCity, schState, schZip)

values (3000, 'Brownwood', 'Mr. Peanut', '(456)123-4567', 'Wizard', 'Miami', 'Florida', 33101);

insert into enroll(schoolID, studentID)

values (3000, 1005);

insert into enroll(schoolID, studentID)

values (3000, 1006);

insert into enroll(schoolID, studentID)

values (3000, 1007);

insert into public\_school(schoolID, distName)

values (3000, 'FireBomb');

insert into school(schoolID, schName, principalName, principalNum, schStreet, schCity, schState, schZip)

values (1000, 'Lincoln High', 'Dr. Plum', '(789)123-4567', 'Tuna','Malboro', 'Maryland', 20772);

insert into enroll(schoolID, studentID)

values (1000, 1008);

insert into enroll(schoolID, studentID)

values (1000, 1009);

insert into enroll(schoolID, studentID)

values (1000, 1010);

insert into private\_school(schoolID, tuition, affiliation)

values (1000, 8000, 'Baptist');

insert into school(schoolID, schName, principalName, principalNum, schStreet, schCity, schState, schZip)

values (7000, 'Pebble Throw', 'Mrs. Dagger', '(903)123-4567', 'Shoftner', 'Longview', 'Texas', 75604);

insert into enroll(schoolID, studentID)

values (7000, 1003);

insert into enroll(schoolID, studentID)

values (7000, 1004);

insert into enroll(schoolID, studentID)

values (7000, 1022);

insert into public\_school(schoolID, distName)

values (7000, 'PTID');

insert into school(schoolID, schName, principalName, principalNum, schStreet, schCity, schState, schZip)

values (4000, 'Springfield High', 'Mr. Skinner', '(636)123-4888', 'Happy', 'Springfield', 'Missouri', 65800);

insert into enroll(schoolID, studentID)

values (4000, 1018);

insert into enroll(schoolID, studentID)

values (4000, 1019);

insert into enroll(schoolID, studentID)

values (4000, 1020);

insert into public\_school(schoolID, distName)

values (4000, 'SID');

insert into school(schoolID, schName, principalName, principalNum, schStreet, schCity, schState, schZip)

values (5000, 'Wayne High', 'Mr. Wayne', '(555)123-4888', 'Bat Wing', 'Gotham', 'New Jersey', 65800);

insert into enroll(schoolID, studentID)

values (5000, 1014);

insert into enroll(schoolID, studentID)

values (5000, 1015);

insert into public\_school(schoolID, distName)

values (5000, 'Justice League');

insert into school(schoolID, schName, principalName, principalNum, schStreet, schCity, schState, schZip)

values (6000, 'Globe Trotter', 'Mr. Smithers', '(543)123-4888', 'Table Cloth', 'Frisco', 'Texas', 75033);

insert into enroll(schoolID, studentID)

values (6000, 5557);

insert into enroll(schoolID, studentID)

values (6000, 5558);

insert into enroll(schoolID, studentID)

values (6000, 5559);

insert into public\_school(schoolID, distName)

values (6000, 'Power Ranger');

commit;

**Query 3**

insert into exam(subject, year, minPassScore, dateGiven)

values ('Math', 2017, 65, '2/may/2017');

insert into takes(studentID, subject, year, score, testLocation)

values (1001, 'Math', 2017, 50, 'Texas');

insert into takes(studentID, subject, year, score, testLocation)

values (1021, 'Math', 2017, 85, 'Texas');

insert into takes(studentID, subject, year, score, testLocation)

values (1002, 'Math', 2017, 55, 'Texas');

insert into takes(studentID, subject, year, score, testLocation)

values (1003, 'Math', 2017, 85, 'Texas');

insert into takes(studentID, subject, year, score, testLocation)

values (1004, 'Math', 2017, 80, 'Texas');

insert into takes(studentID, subject, year, score, testLocation)

values (1005, 'Math', 2017, 92, 'Texas');

insert into takes(studentID, subject, year, score, testLocation)

values (1020, 'Math', 2017, 85, 'Texas');

insert into exam(subject, year, minPassScore, dateGiven)

values ('Math', 2016, 65, '2/may/2016');

insert into takes(studentID, subject, year, score, testLocation)

values (1001, 'Math', 2016, 35, 'Texas');

insert into takes(studentID, subject, year, score, testLocation)

values (1021, 'Math', 2016, 44, 'Texas');

insert into takes(studentID, subject, year, score, testLocation)

values (1002, 'Math', 2016, 35, 'Texas');

insert into takes(studentID, subject, year, score, testLocation)

values (1003, 'Math', 2016, 64, 'Texas');

insert into takes(studentID, subject, year, score, testLocation)

values (1004, 'Math', 2016, 85, 'Texas');

insert into takes(studentID, subject, year, score, testLocation)

values (1005, 'Math', 2016, 85, 'Texas');

insert into takes(studentID, subject, year, score, testLocation)

values (1020, 'Math', 2016, 73, 'Texas');

insert into exam(subject, year, minPassScore, dateGiven)

values ('Chemistry', 2017, 65, '2/february/2017');

insert into takes(studentID, subject, year, score, testLocation)

values (1006, 'Chemistry', 2017, 98, 'Florida');

insert into takes(studentID, subject, year, score, testLocation)

values (1007, 'Chemistry', 2017, 45, 'Florida');

insert into takes(studentID, subject, year, score, testLocation)

values (1008, 'Chemistry', 2017, 100, 'Florida');

insert into takes(studentID, subject, year, score, testLocation)

values (1009, 'Chemistry', 2017, 90, 'Florida');

insert into takes(studentID, subject, year, score, testLocation)

values (1010, 'Chemistry', 2017, 97, 'Florida');

insert into exam(subject, year, minPassScore, dateGiven)

values ('Geography', 2016, 65, '25/december/2016');

insert into takes(studentID, subject, year, score, testLocation)

values (1016, 'Geography', 2016, 98, 'Maryland');

insert into takes(studentID, subject, year, score, testLocation)

values (1017, 'Geography', 2016, 45, 'Maryland');

insert into takes(studentID, subject, year, score, testLocation)

values (1018, 'Geography', 2016, 100, 'Maryland');

insert into takes(studentID, subject, year, score, testLocation)

values (1019, 'Geography', 2016, 90, 'Maryland');

insert into takes(studentID, subject, year, score, testLocation)

values (1022, 'Geography', 2016, 97, 'Maryland');

insert into exam(subject, year, minPassScore, dateGiven)

values ('Writing', 2016, 65, '25/november/2016');

insert into takes(studentID, subject, year, score, testLocation)

values (1011, 'Writing', 2016, 98, 'California');

insert into takes(studentID, subject, year, score, testLocation)

values (1012, 'Writing', 2016, 45, 'California');

insert into takes(studentID, subject, year, score, testLocation)

values (1013, 'Writing', 2016, 100, 'California');

insert into takes(studentID, subject, year, score, testLocation)

values (1014, 'Writing', 2016, 90, 'California');

insert into takes(studentID, subject, year, score, testLocation)

values (1015, 'Writing', 2016, 97, 'California');

insert into exam(subject, year, minPassScore, dateGiven)

values ('History', 2017, 60, '2/april/2017');

insert into takes(studentID, subject, year, score, testLocation)

values (1001, 'History', 2017, 90, 'New Jersey');

insert into takes(studentID, subject, year, score, testLocation)

values (1002, 'History', 2017, 75, 'New Jersey');

insert into takes(studentID, subject, year, score, testLocation)

values (1003, 'History', 2017, 95, 'New Jersey');

insert into takes(studentID, subject, year, score, testLocation)

values (1004, 'History', 2017, 75, 'New Jersey');

insert into takes(studentID, subject, year, score, testLocation)

values (1005, 'History', 2017, 95, 'New Jersey');

insert into takes(studentID, subject, year, score, testLocation)

values (1020, 'History', 2017, 65, 'New Jersey');

insert into exam(subject, year, minPassScore, dateGiven)

values ('Science', 2017, 65, '1/january/2017');

insert into takes(studentID, subject, year, score, testLocation)

values (1001, 'Science', 2017, 98, 'Texas');

insert into takes(studentID, subject, year, score, testLocation)

values (1021, 'Science', 2017, 77, 'Texas');

insert into takes(studentID, subject, year, score, testLocation)

values (5557, 'Science', 2017, 98, 'Texas');

insert into takes(studentID, subject, year, score, testLocation)

values (5558, 'Science', 2017, 47, 'Texas');

insert into takes(studentID, subject, year, score, testLocation)

values (5559, 'Science', 2017, 60, 'Texas');

insert into takes(studentID, subject, year, score, testLocation)

values (5557, 'Math', 2017, 98, 'Texas');

insert into takes(studentID, subject, year, score, testLocation)

values (5558, 'Math', 2017, 47, 'Texas');

insert into takes(studentID, subject, year, score, testLocation)

values (5559, 'Math', 2017, 60, 'Texas');

insert into takes(studentID, subject, year, score, testLocation)

values (1005, 'Science', 2017, 98, 'Texas');

insert into takes(studentID, subject, year, score, testLocation)

values (1006, 'Science', 2017, 47, 'Texas');

insert into takes(studentID, subject, year, score, testLocation)

values (1007, 'Science', 2017, 60, 'Texas');

commit;

**Query 4**

insert into multChoice\_question(year, subject, qNum, correctAns)

values (2017, 'Math', 1, 'A');

insert into multChoice\_question(year, subject, qNum, correctAns)

values (2017, 'Math', 2, 'B');

insert into multChoice\_question(year, subject, qNum, correctAns)

values (2017, 'Math', 3, 'B');

insert into multChoice\_question(year, subject, qNum, correctAns)

values (2017, 'Math', 4, 'B');

insert into essay\_question(year, subject, qNum, collIdeas, expAnsLenLines)

values (2017, 'Math', 5, 'Induction/Point-Wise', 50);

insert into multChoice\_question(year, subject, qNum, correctAns)

values (2017, 'Math', 6, 'B');

insert into question\_topic(year, subject, qNum, topic)

values (2017, 'Math', 1, 'Algebra');

insert into question\_topic(year, subject, qNum, topic)

values (2017, 'Math', 2, 'Algebra');

insert into question\_topic(year, subject, qNum, topic)

values (2017, 'Math', 3, 'Algebra');

insert into question\_topic(year, subject, qNum, topic)

values (2017, 'Math', 4, 'Algebra');

insert into question\_topic(year, subject, qNum, topic)

values (2017, 'Math', 5, 'Algebra');

insert into question\_topic(year, subject, qNum, topic)

values (2017, 'Math', 6, 'Algebra');

insert into multChoice\_question(year, subject, qNum, correctAns)

values (2017, 'Chemistry', 1, 'D');

insert into multChoice\_question(year, subject, qNum, correctAns)

values (2017, 'Chemistry', 2, 'D');

insert into multChoice\_question(year, subject, qNum, correctAns)

values (2017, 'Chemistry', 3, 'D');

insert into multChoice\_question(year, subject, qNum, correctAns)

values (2017, 'Chemistry', 4, 'B');

insert into essay\_question(year, subject, qNum, collIdeas, expAnsLenLines)

values (2017, 'Chemistry', 5, 'Inorganic/Organic', 35);

insert into question\_topic(year, subject, qNum, topic)

values (2017, 'Chemistry', 1, 'Organic');

insert into question\_topic(year, subject, qNum, topic)

values (2017, 'Chemistry', 2, 'Organic');

insert into question\_topic(year, subject, qNum, topic)

values (2017, 'Chemistry', 3, 'Organic');

insert into question\_topic(year, subject, qNum, topic)

values (2017, 'Chemistry', 4, 'Organic');

insert into question\_topic(year, subject, qNum, topic)

values (2017, 'Chemistry', 5, 'Organic');

insert into multChoice\_question(year, subject, qNum, correctAns)

values (2016, 'Geography', 1, 'A');

insert into multChoice\_question(year, subject, qNum, correctAns)

values (2016, 'Geography', 2, 'D');

insert into multChoice\_question(year, subject, qNum, correctAns)

values (2016, 'Geography', 3, 'A');

insert into multChoice\_question(year, subject, qNum, correctAns)

values (2016, 'Geography', 4, 'B');

insert into essay\_question(year, subject, qNum, collIdeas, expAnsLenLines)

values (2016, 'Geography', 5, 'Asia Countries', 120);

insert into question\_topic(year, subject, qNum, topic)

values (2016, 'Geography', 1, 'East Asia');

insert into question\_topic(year, subject, qNum, topic)

values (2016, 'Geography', 2, 'East Asia');

insert into question\_topic(year, subject, qNum, topic)

values (2016, 'Geography', 3, 'East Asia');

insert into question\_topic(year, subject, qNum, topic)

values (2016, 'Geography', 4, 'East Asia');

insert into question\_topic(year, subject, qNum, topic)

values (2016, 'Geography', 5, 'East Asia');

insert into multChoice\_question(year, subject, qNum, correctAns)

values (2016, 'Writing', 1, 'A');

insert into multChoice\_question(year, subject, qNum, correctAns)

values (2016, 'Writing', 2, 'D');

insert into multChoice\_question(year, subject, qNum, correctAns)

values (2016, 'Writing', 3, 'A');

insert into essay\_question(year, subject, qNum, collIdeas, expAnsLenLines)

values (2016, 'Writing', 4, 'Writing/Compr', 200);

insert into essay\_question(year, subject, qNum, collIdeas, expAnsLenLines)

values (2016, 'Writing', 5, 'Comprehension', 200);

insert into question\_topic(year, subject, qNum, topic)

values (2016, 'Writing', 1, 'Comprehension');

insert into question\_topic(year, subject, qNum, topic)

values (2016, 'Writing', 2, 'Comprehension');

insert into question\_topic(year, subject, qNum, topic)

values (2016, 'Writing', 3, 'Comprehension');

insert into question\_topic(year, subject, qNum, topic)

values (2016, 'Writing', 4, 'Comprehension');

insert into question\_topic(year, subject, qNum, topic)

values (2016, 'Writing', 5, 'Comprehension');

insert into multChoice\_question(year, subject, qNum, correctAns)

values (2017, 'History', 1, 'A');

insert into multChoice\_question(year, subject, qNum, correctAns)

values (2017, 'History', 2, 'B');

insert into multChoice\_question(year, subject, qNum, correctAns)

values (2017, 'History', 3, 'B');

insert into multChoice\_question(year, subject, qNum, correctAns)

values (2017, 'History', 4, 'B');

insert into essay\_question(year, subject, qNum, collIdeas, expAnsLenLines)

values (2017, 'History', 5, 'War/Peace', 200);

insert into question\_topic(year, subject, qNum, topic)

values (2017, 'History', 1, 'WWII');

insert into question\_topic(year, subject, qNum, topic)

values (2017, 'History', 2, 'WWII');

insert into question\_topic(year, subject, qNum, topic)

values (2017, 'History', 3, 'WWII');

insert into question\_topic(year, subject, qNum, topic)

values (2017, 'History', 4, 'WWII');

insert into question\_topic(year, subject, qNum, topic)

values (2017, 'History', 5, 'WWII');

insert into multChoice\_question(year, subject, qNum, correctAns)

values (2017, 'Science', 1, 'D');

insert into multChoice\_question(year, subject, qNum, correctAns)

values (2017, 'Science', 2, 'D');

insert into multChoice\_question(year, subject, qNum, correctAns)

values (2017, 'Science', 3, 'D');

commit;

**Query 5**

insert into expert(expertID, expSuffix, expTitle, expLast, expMidIn, expFirst, expStreet, expCity, expState, expZip)

values (111, 'Jr.', 'Dr.', 'Pistachio', 'A', 'Tony', 'Down', 'Tyler', 'Texas', 75701);

insert into expert\_expertise(expertID, expertise)

values (111, 'Topology');

insert into expert\_expertise(expertID, expertise)

values (111, 'Algebra');

insert into expert\_expertise(expertID, expertise)

values (111, 'Proofs');

insert into expert\_evaluates(expertID, year, qNum, subject)

values (111, 2017, 5, 'Math');

insert into expert(expertID, expSuffix, expTitle, expLast, expMidIn, expFirst, expStreet, expCity, expState, expZip)

values (222, 'III', 'Mr.', 'Tiger', 'A', 'Sam', 'Up', 'Miami', 'Florida', 33101);

insert into expert\_expertise(expertID, expertise)

values (222, 'Biology');

insert into expert\_expertise(expertID, expertise)

values (222, 'Chemistry');

insert into expert\_evaluates(expertID, year, qNum, subject)

values (222, 2017, 5, 'Chemistry');

insert into expert(expertID, expSuffix, expTitle, expLast, expMidIn, expFirst, expStreet, expCity, expState, expZip)

values (333, null, 'Mrs.', 'Pretzel', 'R', 'Tina', 'Left', 'Longview', 'Texas', 75604);

insert into expert\_expertise(expertID, expertise)

values (333, 'Algebra');

insert into expert\_evaluates(expertID, year, qNum, subject)

values (333, 2016, 5, 'Geography');

insert into expert(expertID, expSuffix, expTitle, expLast, expMidIn, expFirst, expStreet, expCity, expState, expZip)

values (444, 'Sr.', 'Mr.', 'Rozzli', 'Q', 'Rojilio', 'Right', 'Malboro', 'Maryland', 20772);

insert into expert\_expertise(expertID, expertise)

values (444, 'Algebra');

insert into expert\_evaluates(expertID, year, qNum, subject)

values (444, 2016, 4, 'Writing');

insert into expert(expertID, expSuffix, expTitle, expLast, expMidIn, expFirst, expStreet, expCity, expState, expZip)

values (555, null, 'Mr.', 'Simpson', 'A', 'Homer', 'Springfield', 'East', 'Missouri', 65800);

insert into expert\_expertise(expertID, expertise)

values (555, 'World History');

insert into expert\_evaluates(expertID, year, qNum, subject)

values (555, 2017, 5, 'History');

insert into expert(expertID, expSuffix, expTitle, expLast, expMidIn, expFirst, expStreet, expCity, expState, expZip)

values (585, null, 'Mr.', 'Waternozze', 'R', 'Apollo', 'Edwin', 'Rocky', 'Phiwick', 98765);

insert into expert\_expertise(expertID, expertise)

values (585, 'World History');

insert into expert\_evaluates(expertID, year, qNum, subject)

values (585, 2017, 5, 'History');

commit;

**Query 6**

insert into expert(expertID, expSuffix, expTitle, expLast, expMidIn, expFirst, expStreet, expCity, expState, expZip)

values (666, null, 'Mr.', 'Wayne', 'R', 'Tibberus', 'Maple', 'Gotham', 'New Jersey', 98765);

insert into expert\_expertise(expertID, expertise)

values (666, 'Crime');

insert into expert\_expertise(expertID, expertise)

values (666, 'Ethics');

insert into expert\_leads(expertID, year, qNum, subject)

values (666, 2017, 5, 'Math');

insert into expert(expertID, expSuffix, expTitle, expLast, expMidIn, expFirst, expStreet, expCity, expState, expZip)

values (777, null, 'Mr.', 'Banner', 'R', 'Bruce', 'North', 'Gotham', 'New Jersey', 98765);

insert into expert\_expertise(expertID, expertise)

values (777, 'Sociology');

insert into expert\_leads(expertID, year, qNum, subject)

values (777, 2017, 5, 'Chemistry');

insert into expert(expertID, expSuffix, expTitle, expLast, expMidIn, expFirst, expStreet, expCity, expState, expZip)

values (888, null, 'Mrs.', 'Honey', 'R', 'Sue', 'South', 'Longview', 'Texas', 75604);

insert into expert\_expertise(expertID, expertise)

values (888, 'Crime');

insert into expert\_leads(expertID, year, qNum, subject)

values (888, 2016, 5, 'Geography');

insert into expert(expertID, expSuffix, expTitle, expLast, expMidIn, expFirst, expStreet, expCity, expState, expZip)

values (999, 'IV', 'Dr.', 'Sandler', 'R', 'Adam', 'Duncan', 'Uncertain', 'Texas', 75661);

insert into expert\_expertise(expertID, expertise)

values (999, 'Writing');

insert into expert\_leads(expertID, year, qNum, subject)

values (999, 2016, 5, 'Geography');

insert into expert(expertID, expSuffix, expTitle, expLast, expMidIn, expFirst, expStreet, expCity, expState, expZip)

values (911, 'Jr.', 'Dr.', 'Pepper', 'P', 'Petter', 'Plaza', 'Metropolis', 'Delaware', 22221);

insert into expert\_expertise(expertID, expertise)

values (911, 'History');

insert into expert\_leads(expertID, year, qNum, subject)

values (911, 2017, 5, 'History');

insert into region(state, region)

values ('Texas', 'SouthWest');

insert into region(state, region)

values ('Florida', 'East Coast');

insert into region(state, region)

values ('Maryland', 'Mid-Atlantic');

insert into region(state, region)

values ('Oregon', 'Pacific North');

insert into region(state, region)

values ('Delaware', 'Mid-Atlantic');

insert into region(state, region)

values ('New Jersey', 'Mid-Atlantic');

insert into region(state, region)

values ('Missouri', 'Mid-Atlantic');

commit;

**Query 7**

select subject, studentID, score, stuLevel gradelevel, testlocation

from student natural join takes

where stuState = 'Texas'

and year = 2017

order by gradelevel, score;

**Query 8**

select schName, to\_char((PercentPass \*100), '9999D99') || '%' PassPer

from school, (select passExam.schoolID, (numPass/numAttempt)PercentPass

from passExam, totalExamAttempt

where passExam.schoolID = totalexamattempt.schoolID)x

where school.schoolID = x.schoolID

order by percentpass;

**Query 9**

select schName, to\_char(tuition, '$9,999.99' ) tuition, principalName, principalNum

from private\_school

natural join

(select schoolID, schName, principalName, principalNum

from school

where schoolID in

(select schoolID from

(select schoolID, percentPass

from

(select y.schoolID, (passed/total) percentPass

from (select schoolID, count(\*) passed

from (PrivateSchoolTestScores)

where score >= minpassscore

group by schoolID)x, (select schoolID, count(\*) total

from (PrivateSchoolTestScores)

group by schoolID)y

where y.schoolID = x.schoolID)

where percentpass =

(select max(percentpass)

from

(select y.schoolID, (passed/total) percentPass

from (select schoolID, count(\*) passed

from (PrivateSchoolTestScores)

where score >= minpassscore

group by schoolID)x, (select schoolID, count(\*) total

from (PrivateSchoolTestScores)

group by schoolID)y

where y.schoolID = x.schoolID)))));

**Query 10**

select region

from

(select y.region, (numhs/total) percentHome

from

(select region, count(homeschooled) numhs

from turbo

where homeschooled = 'yes'

group by region)x,

(select region, count(region) total

from turbo

group by region)y

where x.region = y.region)

where percenthome = (select maxp from

(select max(percenthome) maxp

from(

select z.region, (numhs/total) percentHome

from

(select region, count(homeschooled) numhs

from turbo

where homeschooled = 'yes'

group by region)r,

(select region, count(region) total

from turbo

group by region)z

where r.region = z.region)));

**Query 11**

select expLast, expStreet, expCity, expState, expZip

from expert

where expertID in

(select expertID from expert\_evaluates

where subject in

(select subject

from EssayExamPassPercent

where percentpass in

(select max(percentpass)

from EssayExamPassPercent))

and year in

(select year

from EssayExamPassPercent

where percentpass in

(select max(percentpass)

from EssayExamPassPercent)))

or expertID in

(select expertID from expert\_leads

where subject in

(select subject

from EssayExamPassPercent

where percentpass in

(select max(percentpass)

from EssayExamPassPercent))

and year in

(select year

from EssayExamPassPercent

where percentpass in

(select max(percentpass)

from EssayExamPassPercent)))

order by expLast;

**Query 12**

select schName

from school

where schoolID in

(select schoolID

from percentChange

where dif in (select max(dif) from percentchange));

**Query 13**

select expTitle, expLast

from expert, expert\_expertise

where expert.expertID = expert\_expertise.expertID

AND expertise = 'World History'

order by expLast;

**Query 14**

select testLocation, count(testLocation) as NumTimesUsed

from

(select distinct takes.subject, takes.year, testLocation

from exam, takes

where exam.subject = takes.subject AND

exam.year = takes.year)

group by testLocation;

**Query 15**

select topic, nummultcqs, numessayqs

from essayqtopics natural join multchqtopics;

**Query 16**

select subject, Year, score

from student natural join takes natural join exam

natural join expert\_leads natural join expert

where stuZip = expZip;

**Query 17**

update

exam

set

minPassScore = minPassScore + 5

where subject = 'Science' and year IN

(select exam.year

from exam, multChoice\_question

where exam.subject = multChoice\_question.subject AND

exam.year = multChoice\_question.year AND exam.subject = 'Science');

**Query 18**

update

private\_school

set

tuition = tuition \* 1.10

where

schoolID not in

(select private\_school.schoolID

from private\_school, public\_school

where private\_school.schoolID = public\_school.schoolID);

**Query 19**

update

private\_school

set

tuition = tuition \* 1.10

where

schoolID not in

(select private\_school.schoolID

from private\_school, public\_school

where private\_school.schoolID = public\_school.schoolID);

**Query 20**

delete from enroll

where schoolID in

(select schoolID

from statesciencepassrate

where (schstate, percentpass) in

(select schstate, min(percentpass) percentpass

from statesciencepassrate

group by schstate));

delete from school

where schoolID in

(select schoolID

from statesciencepassrate

where (schstate, percentpass) in

(select schstate, min(percentpass) percentpass

from statesciencepassrate

group by schstate));