RIT | Golisano College of Computing and Information Sciences
School of Information

ISTE-722 Database Connectivity and Access
Practical 1

Name: _____

For this practical exam, you may use any non-human reference that your classmates would also have access to. This includes anything from our course and anything you might find on the Internet. Submit your Java files to the Practical 1 drop box. If you are reading this, the clock has started and you have 50 minutes to make your submission. Don't get caught without having submitted your code, make a submission each time you reach a milestone. I will grade ONLY your last submission of the two required Java files.

Be sure to download and run the AOMSimple.sql script. Also be sure to download the DBUtility.jar, mysql-connector-java-8.0.20.jar, and the AOMSimple.properties file. In the properties file, adjust the property values to reflect your credentials and local configuration.

RIT | Golisano College of Computing and Information Sciences
School of Information

There are two Java starter files that you must complete. One is named Isolate.java and is a row-oriented DAO; the other is named Isolates.java and is a table-oriented DAO. You should not change anything that is already in these files.

Following are details about the requirements:

Isolate.java

- I have provided attributes, constructors, accessors, mutators, and a main method for testing purposes. Make no changes to this code.
- I have provided signatures for the 4 basic CRUD methods. Your task is to write appropriate code for each of these methods so that they function properly.

Isolates.java

- I have provided attributes, constructors, and accessors. Make no changes to this code.
- I have not provided any mutators—none are required.
- I have not provided a main method for testing purposes because your implementation of the remaining code will vary and therefore require a custom testing method. You are not required to include a main method, but may do so if you wish.
- Your one requirement for this class is to implement a read method that will:
    1. Allow the caller to specify any number of filtering criteria. You only need to consider exact matches, e.g., name = 'Patel'. (I recommend that you use a 2-d List for this, but you are free to use any reasonable approach).
    2. Build and execute the appropriate SQL statement(s).
    3. Populate the class attribute.

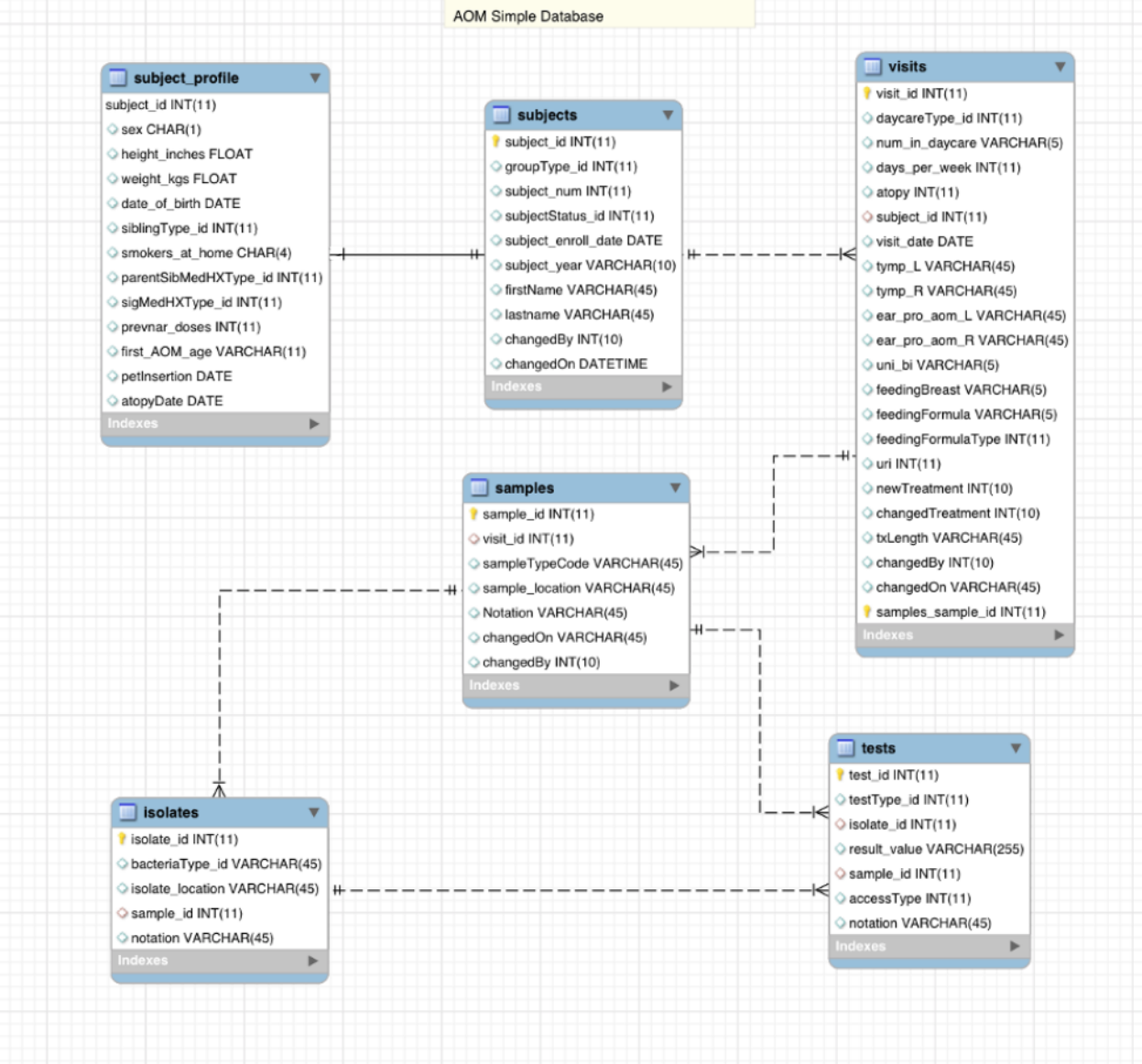
Following is a list of methods to be completed and how many points you'll receive for each one. You may write additional methods to support your work, but this is not required

Class	Method	Points
Isolate	insert	10
	read	8
	update	8
	delete	8
Isolates	read	16
		50

When grading your work, I will look for:

- Proper functionality
- Proper use of prepared statements, stored procedures, and any other constructs we have covered in class
- Utilization of proper coding methods
- Reasonable variable names used
- Code is easy to follow and understand

Technically, comments are optional, but if you want partial credit, I need to be able to understand what you are doing and the burden for that is on you.


AOM Simple Database
subject_profile
subject_id INT(11)
sex CHAR(1)
height_inches FLOAT
weight_kgs FLOAT
date_of_birth DATE
siblingType_id INT(11)
smokers_at_home CHAR(4)
parentSibMedHXType_id INT(11)
sigMedHXType_id INT(11)
prevnar_doses INT(11)
first_AOM_age VARCHAR(11)
petInsertion DATE
atopyDate DATE
Indexes
subjects
subject_id INT(11)
groupType_id INT(11)
subject_num INT(11)
subjectStatus_id INT(11)
subject_enroll_date DATE
subject_year VARCHAR(10)
firstName VARCHAR(45)
lastname VARCHAR(45)
changedBy INT(10)
changedOn DATETIME
Indexes
visits
visit_id INT(11)
daycareType_id INT(11)
num_in_daycare VARCHAR(5)
days_per_week INT(11)
atopy INT(11)
subject_id INT(11)
visit_date DATE
tymp_L VARCHAR(45)
tymp_R VARCHAR(45)
ear_pro_aom_L VARCHAR(45)
ear_pro_aom_R VARCHAR(45)
uni_bi VARCHAR(5)
feedingBreast VARCHAR(5)
feedingFormula VARCHAR(5)
feedingFormulaType INT(11)
uri INT(11)
newTreatment INT(10)
changedTreatment INT(10)
txLength VARCHAR(45)
changedBy INT(10)
changedOn VARCHAR(45)
samples_sample_id INT(11)
Indexes
samples
sample_id INT(11)
visit_id INT(11)
sampleTypeCode VARCHAR(45)
sample_location VARCHAR(45)
Notation VARCHAR(45)
changedOn VARCHAR(45)
changedBy INT(10)
Indexes
tests
test_id INT(11)
testType_id INT(11)
isolate_id INT(11)
result_value VARCHAR(255)
sample_id INT(11)
accessType INT(11)
notation VARCHAR(45)
Indexes
isolates
isolate_id INT(11)
bacteriaType_id VARCHAR(45)
isolate_location VARCHAR(45)
sample_id INT(11)
notation VARCHAR(45)
Indexes