University of British Columbia, Vancouver

Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: ____3____

Date: ___01/11/23_____

Group Number: ___126_____

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Tyler Kerswell	52101672	z8v1d	tylerkerswell@gmail.com
Julie Ryu	16183253	t4v5i	julieryu2@gmail.com
Natalia Garcia-Arias	54821806	m1p7e	garciarias.natalia@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

University of British Columbia, Vancouver

Department of Computer Science

Project Description

Project Domain: Art gallery operations and event management

Our database aims to organize data related to various aspects of art gallery operations including event organization, exhibit planning, employee management.
People will be able to manage their art gallery with this database. It will allow managers to keep track of events hosted at their art galleries as well as the artwork and exhibits they showcase there.

MILESTONE 4 BREAKDOWN: due Dec 1, 2023 10:00 PM

NOTE: we cannot change the code between Milestone 4 and Milestone 5, so we must meet all demo requirements when we submit Milestone 4

Deliverable	Status	Notes	
1.Cover Page	Not …		
A single SQL script that can be used to create all the tables and database	Not …	• If using multiple scripts, make sure to concatenate them and hand in only a SINGLE SQL script ◦ Make sure that queries have some non-trivial answers (same thing with aggregation) ◦ Need to have groups that have more than one row ◦ Script should be runnable	
THIS STUFF SHOULD BE IN A PDF FILE:	Not …		
Short project description and what is accomplished	Not …		

University of British Columbia, Vancouver

Department of Computer Science

Deliverable	Status	Notes	
Description of how your final schema differed from the schema you turned in	Not …	• If it is different, explain why	
A copy of the schema and screenshots that show what data is present in each relation ter the SQL script from item #2 is run	Not …	• DO not use DDL statements for this. List relational schemas with the primary key attributes underlined and foreign keys bolded • This can be done with screenshots from SQL Plus or create a representation of your relational instances through a spreadsheet program • Just make sure to clearly label which relation a given instance refers to	
A list of all SQL queries used and where it can be found in the code (file name, line numbers)	Not …	☐ Check SQL query requirements in the rubric on Canvas:	
Screenshots demonstrating the functionality of each query using the GUI.	Not …	• Demonstrate before/during/after progression of events. • Clearly label each set of screenshots with the name of the query it is meant to address • You only need to include screenshots for the required queries	
README.txt file if there is anything you need to add that is not in the PDF	Not …		

University of British Columbia, Vancouver

Department of Computer Science

Breakdown of SQL Query Requirement **TO IMPLEMENT**

This also includes some other application features.

Query	Status	Notes	Assigned to
GUI	Not started	The GUI doesn't need to be fancy, but at least a basic GUI is necessary. The GUI should be usable from the perspective of a non-computer scientist and should not include SQL syntax. If the project does not have a GUI (e.g., it is run through command line), a penalty of 40% on the total value of the project grade will be applied.	Everyone
USER NOTIFICATION	Not started	The user will receive a success or failure notification upon the completion of an insert, update, delete action and will have a way to verify the action's effect on the database.	Everyone
INSERT	Not started	☐ User should be able to specify what values to insert ☐ Operation should affect more than one table ☐ Should be able to handle the case where the FK value in the tuple does not exists in the table that is being referred to ☐ Tables that the insert operation will run on can be pre-chosen by the group ☐ Need to provide user with success or failure notification upon completion ☐ The **assertions and constraints required** to model the ER diagram have also been included. ☐ **DROP TABLE** statements have been included at the beginning of the file to allow the file to be run multiple times (if needed). All the statements are in a single file.	Natalia
DELETE	Not started	☐ Implement a cascade-on-delete situation (or alternative if DB system doesn't provide this) ☐ Table that this operation will run on can be chosen by the group ☐ Need to provide user with success or failure notification upon completion	Natalia

University of British Columbia, Vancouver

Department of Computer Science

Query	Status	Notes	Assigned to
UPDATE	Not started	☐ User should be able to update any number of non-primary key attributes in a relation ☐ Relation used for this operation must have **at least two non-primary key** attributes. **At least one non-primary key** attribute must have either a UNIQUE constraint or be a foreign key ☐ Application should present the tuples that are available so that the user can select which tuple they want to update ☐ Need to provide user with success or failure notification upon completion	Tyler
SELECTION	Not started	☐ User can specify filter conditions for a given table ☐ User can determine what shows up in the WHERE clause ☐ User should be allowed to search for tuples using any number of AND/OR clauses ☐ Group can choose which table to run this query on	Julie
PROJECTION	Not started	☐ User is able to choose any number of attributes to view from any relation in the database ☐ Non-selected attributes must not appear in the result ☐ One or more tables in relation must contain at least 4 attributes ☐ Application must dynamically load the tables from the database ☐ Don't need to allow projection over multiple relations	Julie
JOIN	Not started	☐ Create one query which joins at least 2 tables and performs a meaningful query ☐ Provide an interface for the user to execute this query ☐ User must provide at least one value to qualify in the WHERE clause ☐ Group can choose which tables will be affected by the query	Tyler
AGGREGATION WITH	Not started	☐ Create one query that requires the use of aggregation (min, max, average, or count are ok)	Natalia

University of British Columbia, Vancouver

Department of Computer Science

Query	Status	Notes	Assigned to
GROUPBY		☐ Provide an interface (HTML button/dropdown, etc) for the user to execute this query ☐ Schema can be statically set but tuples used in query cannot be predetermined	
AGGREGATION WITH HAVING	Not started	• Create one meaningful query that requires the use of a HAVING clause, and provide an interface for this • Schema can be statically set but the tuples used in query cannot be predetermined	Tyler
NESTED AGGREGATION WITH GROUPBY	Not started	• Create one query that finds some aggregated value for each group • Schema can be statically set but the tuples used in query cannot be predetermined • *Note the difference between this query and the above Aggregation Query. Must use separate distinct queries for this criterion and the Aggregation Query (i.e., do not double dip)*	Julie
DIVISION	Not started	• Create one query of this category and provide an interface (i.e., HTML button, etc.) for the user to execute this query (e.g., find all the customers who bought all the items). • The schema can be statically set but the tuples used in the query cannot be predetermined.	Tyler
Basic Error Handling	Not started	• User receives notifications about user errors ○ Ex: trying to insert a duplicate value, invalid input	Everyone
Security Practices / Sanitization	Not started	• Values from the user are not directly used in the database • Follow basic security practices to prevent injection and rainbow attacks	Everyone

University of British Columbia, Vancouver

Department of Computer Science

MILESTONE 5 TASKS OVERVIEW due Dec 4, 2023 9:00 AM

	Status	Notes	Assigned to
Book to sign up for demo	Not Start…	**NEEDS TO BE DONE BY Nov 24, 2023** **Link to sign up:** https://docs.google.com/spreadsheets/d/1UEChHZRnk8DLW_bW6WgU5kO_xRSoiT0p9Yvi7FV4avA/edit#gid=994582148 **Slots: Dec 4 - Dec 7**	Everyone
Be sure to also have SQL Plus ready so it can be used to double check the result of a given action in your application	Not Start…		Everyone
Should have the commands to do this ready to go so time is not wasted on dealing with constraints due to command order.	Not Start…	a. It is also possible that the TA will ask the group to change the database location to another spot (e.g., the TA's own account).	Everyone
Practice Demo at least once	Not Start…		Everyone

University of British Columbia, Vancouver

Department of Computer Science

Project Timeline

We will first work individually on our respective sections of the database to make the backend. Afterwards, we will work collectively to create the GUI.

Respective Sections

Tyler: Donor, donated by, Artwork, Artist, Features, Exhibition
Natalia: host, Event, participate in, visitor, visit, Curates, supervises
Julie: Employees, Event Staff, Researcher, Curator, works on, Projects, funds

Pre-Implementation Work Nov 3, 2023

- [ ] Meet to design the architecture of our system (after TA meeting)
- [ ] Do any necessary setups
    - [ ] https://www.students.cs.ubc.ca/~cs-304/resources/jdbc-oracle-resources/jdbc-java-setup.html
    - [ ] https://www.students.cs.ubc.ca/~cs-304/resources/jdbc-oracle-resources/jdbc-java-looking-through-code.html#converting-between-java-and-oracle
- [ ] Decide on how to structure transactions

Back-end

Note: This is based off the structure from this sample project:
https://github.students.cs.ubc.ca/CPSC304/CPSC304_Java_Project

- [ ] SQL script to create tables and instances due: Nov 8, 2023
- [ ] SQL queries due Nov 8, 2023
- [ ] Write java code needed to embed database system in our application Nov 15, 2023
- [ ] Connect our implementation to database system Nov 15, 2023

Front-end

- [ ] Implement GUI for queries due Nov 29, 2023
    - [ ] Login Window
    - [ ] Search bar to search for specific artists, artworks, exhibitions, and events

University of British Columbia, Vancouver

Department of Computer Science

- [ ] Fund option button that allows users to either donate an artwork or fund a project
- [ ] …More to be added