

Any suggestion

Your Resume is Ready!

Rank	名字	图片	link
1			
...			
5			

✓

login ✓	username	<input type="text"/>
	password	<input type="text"/>
	<div>Register Submit</div>	

login ✓	username	<input type="text" value="test"/>
	password	<input type="text" value="test"/>
	<div>Register Submit ✓</div>	

"Register" table create

✓

login ✓	username	<input type="text"/>	(Not null, 不重复)
	Email	<input type="text"/>	
	phone number	<input type="text"/>	
	password	<input type="text"/>	(Not null, 不重复)
			<div>Submit</div>

✓

=

6

Store

logout

art style

rank

platform

Submit

Your Recommendation is Ready!

	名字	Episode	publish data	Genre

表

	username <input type="text" value="admin"/>
	password <input type="text" value="admin"/>
login	

delete	Anime name <input type="text"/> (not null)
update	publish data <input type="text"/>
	episode <input type="text"/>
insert	Genre <input type="text"/>

↑
Admin query

2020. 11.18

Homepage

Register

Login Account

Username

Password

1. Insert / delete / update

- Anime name
- publish date
- episode
- Genre

2. Selection (Personalized Rank)

- Anime Rank

top 40

top 45

top 50

top 1000

3. Projection (Personalized Anime)

- Broadcast Platform

Youtube

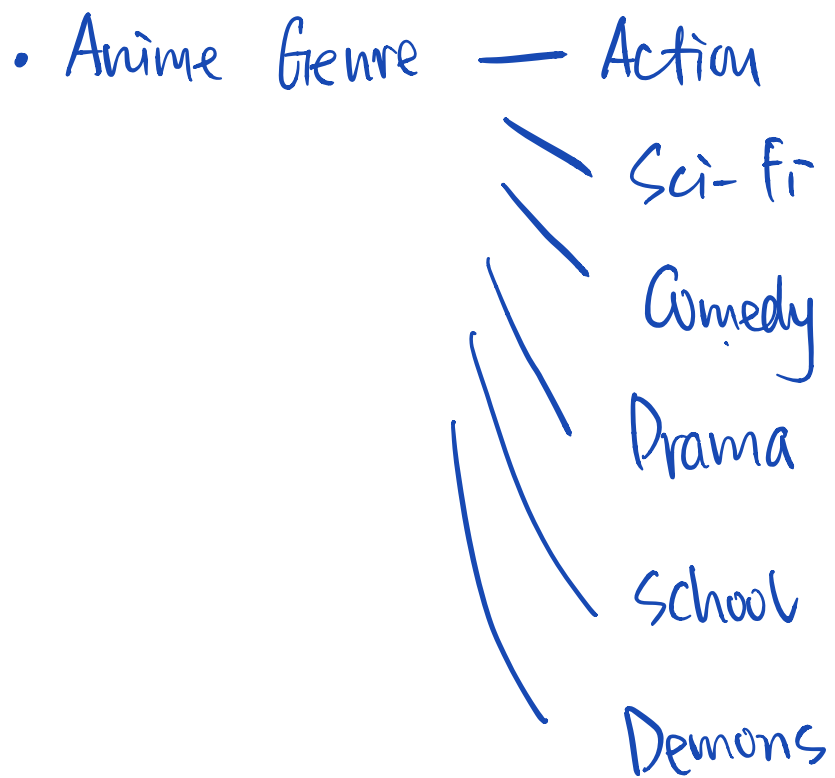
Bilibili

Netflix

Amazon Prime

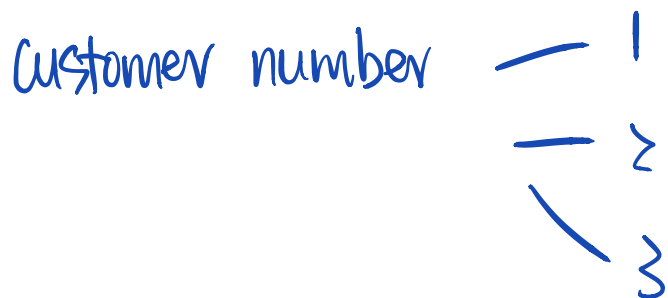
Funimation

Google Play



4. Join

Find the price and name for the merchandise for



Personalized Rank (Selection)

Select your anime in

top 40
top 45
top 50
top 100

Submit

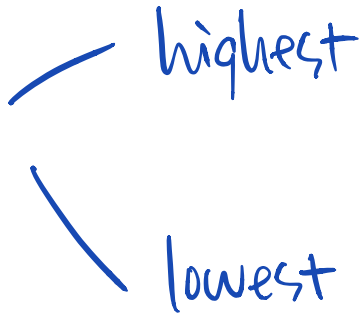
Personalized Anime

Select your anime with

- Platform
 - Genre
 - Rank Type
- } dropdowns

Submit


5. Group By (Top Ranked)

Find the  highest ranked anime in
lowest

Our database (default: rank by popularity)

6. Having

Find the count of Greater type greater

than  3
2
1

7. Nested Group By

Find category of the merchandise where average price
is the same as max price over all category

8. Division

Find all platforms contain all anime

CPSC 304 Project Instructions: Project Final Deliverables

Milestone 3: Comments about Deliverables and Marking

Updated: November 13, 2020 @ 14:05

The marking scheme is tentatively out of 16, although it will be weighted to 40% of the overall project grade at the end of the course. The checklist described below echoes what we've already written up in the Project Specifications (see the Project Description document that you're all familiar with).

After you submit your work on Canvas, you can update a prior submission on Canvas until the deadline of Friday at 23:59 on November 20, 2020. Note the 25% penalty per day late.

After the deadline, your project work will be frozen, meaning you cannot add/delete things between then and your demo.

Make sure that one of your group members submits the work (just like for previous milestones). Confirm with your other group members when you've submitted the work. Don't assume that someone else from your group is going to submit it.

Screenshots: To give the TAs evidence that your code works, we're requiring you to provide screenshots of your output (including before/after screenshots of updates made following an INSERT operation, for example). This will not only help the TAs mark your code, but it will reduce the stress and risk-factor associated with a demo that doesn't go as smoothly as you would like. The demo has 3 main purposes: (a) a confirmation of the work that you've handed in; (b) a presentation of your group's work, similar to a presentation/demo that you might give in a future job, and it lets the TA see your work in action—so these are good “soft skills” to develop and have; and (c) a chance for your TA to ask questions to your group. In the event that a TA cannot get to all your deliverables during the demo, don't worry, the screenshots and code that you've handed in, will be useful.

Files that your group must include in your **Canvas submission** are as follows:



- Cover page
- Source code for the project
 - Zip up your code.
 - Include the source code that you've written, including the embedded SQL. Don't hand in libraries and other baggage, like large executable files, as this will just create a huge zip file. Basically, hand in what your TA needs to see (your source code, SQL, possibly HTML (especially for PHP), possibly a README file). You can ask your TA, if you're unsure about including something.

- Submit your code to Canvas, even if you used GitHub, etc. The zip format will make it convenient for a TA.
- SQL to create and load the tables
- If the SQL was embedded in your source code, and is clearly identified, you don't have to submit a separate list of your SQL queries.
- A PDF file giving a short description of your project from a business perspective (purpose, targeted audience, and how it helps the business or its users). Include any explanations about missed deliverables, changed deliverables, changed formal specs (if any), changed E-R diagram or entities or relationships (if any), unexpected challenges, new features (if any) above and beyond your formal specs, etc.
 - It is not necessary that you re-draw your E-R diagram—an explanation describing the changes is fine.
- Important: Include screenshots of sample output from your project.
- [Optional] Lastly, you can include a README.txt file if there's anything you want to add that's not included in your PDF file. (Most students will probably include such comments in their PDF.)

Late submission policy: 25% penalty per day late

Here is the tentative marking scheme (we don't expect it to change, unless we've overlooked something).

Milestone 3 Checklist and Tentative Marking Scheme	Out of	Group Mark
<ul style="list-style-type: none"> Cover Page: <ul style="list-style-type: none"> Deduct 1 point, if the cover page was not submitted. Even if it was not submitted, you still need to ask the students for the cover sheet, since it's their statement that they didn't plagiarize, etc. If it is submitted, make sure it was completed successfully. It's OK if someone didn't list their e-mail address. Otherwise, deduct 0.5 or 1.0, depending on how much is missing. 	1	
<ul style="list-style-type: none"> SQL DDL and Loading of Data <ul style="list-style-type: none"> Verify that the correct SQL DDL statements were submitted. These are the <u>statements</u> that were used to create their database. <ul style="list-style-type: none"> 1 point Verify that the appropriate SQL INSERT statements (or load/import or equivalent) have been submitted. These are needed to create the tables. <ul style="list-style-type: none"> 1 point 	1 1	

<ul style="list-style-type: none"> • Documentation <ul style="list-style-type: none"> • How good was the overall documentation for the code, and any external documents or files describing their project? It should include a description of the final project and what the project accomplished. If any changes were made between the previous milestones' specs (like a different schema), that should be noted here, as well. <ul style="list-style-type: none"> • Award 1 point for reasonably good documentation. • Award 0.5 for so-so documentation. • Award 0 points for poor documentation. 	1	
<ul style="list-style-type: none"> • The code and the SQL queries will be evaluated in the next section. Students can provide a list of their SQL used, or they can make it clear in the code provided. 		
 Checklist of Queries. Students need to implement these items, and hand in their implemented code, etc., that they created for these checklist items. Screenshots must be provided. The code must work. This will be demonstrated by the screenshots and by the students during the demo. Half marks can be provided if something doesn't work, but if a good effort has been made at trying to get it to work (as evidenced by the code).		
GUI – The GUI doesn't need to be fancy, but at least a basic GUI is necessary (e.g., buttons to run a query is adequate). Really fancy GUIs can be awarded a bonus point.	2	
Same Deliverables as Handed In? – During the demo, the group shows the TA that the deliverables haven't changed from what they handed in (e.g., timestamps). This is just about the evidence. (We won't accept changes or new features beyond the deadline.)	0.5	
 Insert Operation – Provide an interface for the user to specify some input for the insert operation.	0.5	
Delete Operation – Implement a cascade-on-delete situation. Provide an interface for the user to specify some input for the deletion operation.	0.5	
Update Operation – Provide an interface for the user to specify some input for the update operation.	1	
Selection – Create one query of this category and provide an interface for the user to specify the values of the selection conditions to be returned. Example: <pre>SELECT ... FROM ... WHERE Field1 = :Var1 AND Field2 > :Var2</pre>	1	
Projection – Create one query of this category, with 3-5 attributes in the projection condition, but not SELECT *. If you wish, you can have the user select the attributes from a drop-down list, but we will accept a hard-coded SELECT statement for the table(s) in question. <ul style="list-style-type: none"> • This can be combined with another step if there are 3-5 attributes in the projection (but not SELECT *). 	1	
Join – Create one query in this category, which joins at least 2 tables and performs a meaningful query, and provide an interface for the user to execute this query. The user must provide at least one value to qualify in the	1	

WHERE clause (e.g. join the Customer and the Transaction table to find the names and phone numbers of all customers who have purchased a specific item).		
Aggregation with Group By – Create one query that requires the use of distinct aggregation (<i>min</i> , <i>max</i> , <i>average</i> , or <i>count</i> are all fine), and provide an interface (e.g., HTML button/dropdown, etc.) for the user to execute this query.	1	
Aggregation with Having – Create one meaningful query that requires the use of a HAVING clause, and provide an interface (e.g., HTML button/dropdown, etc.) for the user to execute this query.	1	
Nested Aggregation with Group By – Create one query that finds some aggregated value <i>for each group</i> (e.g., the average number of items purchased per customer). Note the difference between this query and the above Aggregation Query. You must use separate distinct queries for this criterion and the Aggregation Query (i.e., do not double dip). Note that “nested” does not mean, for example, <i>min(max(price))</i> ; instead, it is an aggregation with a Group By as explained in lecture slides.	1.5	
Division – 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).	1	
Unofficial GROUP MARK	Out of 16	
Late Marks <ul style="list-style-type: none"> • Subtract 25% (4.0 marks) per day late 		
Bonus Mark(s) – Create unique features for your application that clearly go over and above what was expected. Some examples include a really fancy GUI, implementing useful trigger(s) and having application logic that deals with it, produces special reports, has a “wow” factor, etc. Keep in mind these bonus points have a HIGH standard to meet. Not all groups will get bonus marks; possibly only a few will get 2 bonus marks (1 mark for each feature).		
Milestone 3 GROUP MARK	Out of 16	