

CS209A Tutorial Week 9

This week you need to implement a simple Grade Query system. We will read data from CSV file [StudentsGrade.csv](#) and then store the data to database.

In general, it is a good design practice if one can separate the parts of the system that include application functions (here processing the grades and presenting them) from the storage functions (here either an output stream to file or a SQLite database). In the real world this would allow you to switch between different databases, network storage locations, or other storage implementations without changing the application itself. This is called modular design and requires planning before you start coding...

Summary of the requirements for this practical:

When the program starts:

- (1) First, you should create an SQLite database that is titled [StudentDB.sqlite](#);
- (2) You should create a table in the database named [StudentsGrade](#), which has two columns [name\(string\)](#) and [grade\(int\)](#).
- (3) you should read the data from [StudentsGrade.csv](#) and insert all the records into the table [StudentsGrade](#) in [StudentDB.sqlite](#).

As you are working through the prac and executing PreparedStatements, you should use the database manager tool such as this Firefox plugin to view the database and check what you have done in Java with JDBC: <https://addons.thunderbird.net/en-us/firefox/addon/sqlite-manager/?src=cb-dl-users>

You should query the database using SQL statements. You need to implement the following SQL commands and PreparedStatements (see the Java database API and lectures):

1. Query a student's grade by name (make the name a parameter)
2. Query the names and grades of students by a grade range between x and y (where x and y are parameters)
3. Query the TOP x of students (where x is a parameter)

The input and output interactions are as follows: (The green font indicates that the user is required to input the command from the console):

1. Query a student's grade by name

Command format: NAME student's name

```
NAME Abebe Veronica
62
```

2. Query the names and grades of students by a grade range

Command format: GRADE down-bound upper-bound

```
GRADE 60 70
Abebe Veronica,62
Acquah Kyra,67
Ando Theodore,67
```

Brantley Van,68
 Chen Ariel,61
 Chen Helen,68
 Contreras Gabriel,67
 Du Claire,68
 Finkle Nathan,61
 Gai Sean,68
 Gitelman Daniel,67
 Guo Jingwen,67
 Huynh Brandon,68
 Johnson Thomas,61
 Joshi Rohan,68
 Kallini Julie,67
 Kim Heidi,67
 Lewandowski Carina,68
 Liotine Camille,61
 Liu Katie,68
 Marcu Theodor,67
 Morck Bobby,68
 Oak Malika,61
 Osgood Morlan,68
 Park Naomi,67
 Sands Spencer,68
 Shoaib Zoya,61
 Sirisakunngam Kwan,68
 Sommers Grace,67
 Stempel Grace,67
 Uberoy Urvashi,68
 Wang Janet,61
 Wedbush Brady,68
 Xu Andrew,67
 Zhao Dora,68

3. Query TOP students

Command format: TOP number

TOP 1
 Afolabi Daniel,100
 Ahmad Fawaz,100
 Ajjarapu Neel,100
 Aravena Scott,100
 Arora Somya,100
 Azam Shamilah,100
 Barkachi Chris,100
 Barnes Kyle,100
 Basili David,100
 Batscha Ziv,100
 Baughman Maggie,100
 Berman Ami,100
 Bethel-Brescia Chaz,100
 Bickenbach Pablo,100
 Catoto Bianca,100
 Chan Charmaine,100
 Chang Anabelle,100
 Chen Alexander,100
 Chen Grace,100
 Cheng Alex,100
 Chou Gene,100
 Cooper Mitchell,100
 Costa Caio,100

Crawford Mariah,100
Cyr Tigar,100
Dai Allen,100
Daniel Noah,100
Dardik Sara,100
Darko Winfred,100
Devraj Arjun,100
Di Vittorio Caroline,100
Ding Kimberly,100
Elgamal Amina,100
Farah Ahmed,100
Fastow Matthew,100
Finch Kevin,100
Freeman Robbie,100
Galstyan Narek,100
Godoy Mateo,100
Goldberg Yoni,100
Goodman Daniel,100
Gupta Arul,100
Guthrie Claire,100
Hack Abby,100
Hamin Maia,100
Hammack Zach,100
He Jacqueline,100
He Nina,100
Hein Anthony,100
Ho Long,100
Hong Grace,100
Houle Brendan,100
Jagetia Rahul,100
Jia Watson,100
Jing Lucy,100
Johnson Nicholas,100
Joseph Rod Eric,100
Jubas Daniel,100
Kane Olivia,100
Kar Nabhonil,100
Katzman Lyra,100
Kim Jason,100
Kim Nicholas,100
Kipkorir Allan,100
Koh Ze-Xin,100
Kohli Aditya,100
Kortum Grace,100
Koslowe Hillel,100
Kostic Aleksandra,100
Lacayo Satuye,100
Laude Audrey,100
Lee Dale,100
Liang Ben,100
Limpasuvan Drew,100
Lin Andy,100
Lin William,100
Liu David,100
Liu Nicholas,100
Lu Christine,100
Markakis Markos,100
Masood Zaynab,100
Mathoniere Alex,100
McDaniel Kolby,100
McElroy David,100

Medghalchi Arman,100
Mendoza Felipe,100
Meng Lillian,100
Miller McGinnis,100
Minge Chris,100
Miron Jack,100
Nganthavee Pang,100
Nguyen Leanna,100
Nishikawa Fu Max,100
O'Leary Natalie,100
Opena Miguel,100
Pachisia Divyanshu,100
Park Raymond,100
Peck Fleming,100
Perez Bryan,100
Phillips Stephen,100
Polcyn Stephen,100
Powell Carlin,100
Rakovitsky Ariel,100
Ramesh Vinay,100
Rao Rohan,100
Ratliffe Joe,100
Raval Hari,100
Ritossa Joel,100
Romano Mensheng,100
Rubin Joseph,100
Schmeller Nicholas,100
Schwartz Uri,100
Shanker Tan,100
Sheng Jenny,100
Sinha Ishan,100
Skepasto Aaron,100
Son Justin,100
Song Yang,100
Spirollari Paskalino,100
Stengel Gabriel,100
Stowe Jarret,100
Suh HJ,100
Sweeny William,100
Sylejmani Doruntina,100
Tam Katie,100
Tan Jason,100
Tang Lauren,100
Thomas Emerson,100
Tian Mason,100
Todd David,100
Vuong Erin,100
Waksbaum Jake,100
Wang Albert,100
Wang Henry,100
Webb Toussaint,100
Weng Angela,100
Wolfe Isaac,100
Xu Jason,100
Xue Ivy,100
Yan Arthur,100
Yap Ben,100
Ye Christopher,100
Yin Emily,100
Ying Karen,100
Yip Suki,100

Yuan Wesley,100 Zaller Isabel,100 Zeng Haley,100 Zou Anne,100 Zuluaga David,100
