CSIS 3475 - DATA STRUCTURES AND ALGORITHMS
JAVA DATABASE PROJECT SPECIFICATION

PREPARED FOR STEPHEN CHIONG, INSTUCTOR

TEAM MEMBERS
Joy Cao (300232405)
Ronnie Hung (300209539)
Matthew Lai (300228124)
Alan Changil Lee (300247998)
Caleb Daesung Park (300221972)

IN THIS DOCUMENT:
A: General Database Rules and Notes
B: DB Commands, Proper Syntax, and Notes on Usage
C: Test Cases

SECTION A: General Database Rules and Notes

Table Definition
Table names and column names cannot have spaces and must be English-language letters. They also cannot be punctuation characters. The string “noJoin” is a reserved keyword and is also not allowed as a table name.
Only the following data types are allowed:

- String (String)
- Integer (int)
- Double (double)

Primary Keys
The first field the user defines when using the CREATE TABLE command MUST be the primary key, which will always be an Integer.
The name of the column must have “PK” in front i.e. “PKcolumnName”.

Foreign Keys
Foreign key definition MUST be done during table creation. The name of the column must have “FK” in front i.e. “FKcolumnName”. Foreign key indicates the primary key of reference table.

List of Reserved Words (tables and columns can’t be named these):

- SELECT
- FROM
- INNER
- JOIN
- NOJOIN
- UPDATE
- IN
- SET
- TO
- DELETE
- ALL
- TABLE
- INSERT
- ORDERBY

Other Database Notes:

- Commands can be written in either lowercase or uppercase.

SECTION B: DB Commands, Proper Syntax, and Notes on Usage

CREATE TABLE Command

```
CREATE TABLE tablename
(
PKcolumn1 int,
column2 datatype,
FKcolumn3 datatype,
...
);
```

- The supported data types are: String, int, double, and date
- The first field/column declared must be the primary key, and will always be of the integer data type.
- Primary keys and foreign keys are declared by having PK or FK in front of the column name. These keys can only be declared during table creation.

example:

```
CREATE TABLE students
(
PKID int,
names String,
FKID int,
);
```

SELECT Command

```
SELECT column1, column2, ... FROM tablename
SELECT column1, column2, ... FROM tablename INNER JOIN joinTablename
```

INNER JOIN Sub-command

Inner Join operation searches foreign key in tablename, and generate temporary table which has all columns and data with connecting foreign key and jointablename.
After inner join, SELECT column, WHERE searching, and ORDERBY sorting work on this generated table.
Eg. SELECT column1OfTablename, column2OfJoinTablename FROM tablename INNER JOIN joinTablename

Limitation: tablename can have only one foreign key and Inner Join supports only one joinTable.
When user select certain columns with INNER JOIN, that culumn names should be indentical.

example:

```
SELECT * FROM students INNER JOIN instructor
```

Result

students

PKID	names	FKID	first_name	last_name
1	Matt	2	Mehwish	Bashir
2	Ronnie	1	Stephen	Chiong
3	Alan	3	Gilbert	Tsui
4	Joy	2	Mehwish	Bashir
5	Park	1	Stephen	Chiong
6	Shaun	3	Gilbert	Tsui
7	Alex	2	Mehwish	Bashir
8	Andrew	1	Stephen	Chiong
9	Kevin	3	Gilbert	Tsui
11	Steve	4	Michael	Wufka
12	Maria	5	Raymond	Yu

WHERE Command
(some select command)
WHERE column1 operator value operator column2 operator value ...
Eg. Column1 >= 1 && columb2 <5

- The user can include as many conditions has they want

----------Search Part Specification ----------
-
Supported Operators

- >, <, =, >=, <=, !=, like (Case Insensitive)
- &&, ||, AND, OR (Case Insensitive)

Description of Operators

- A > B : find records whose A field's value is bigger than B value
- A >= B : find records whose A field's value is bigger than or equal to B value
- A = B : find records whose A field's value is equal to B value
- A < B : find records whose A field's value is smaller than B value
- A <= B : find records whose A field's value is smaller than or equal to B value
- A != B : find records whose A field's value is not equal to B value
- A like B : find records whose A fields string includes string B. It doesn't need single quote to indicate target string.

Operators' Order

- Process ">, <. =, >=, <=, !=, like" operator first, and combine this result with "&&, ||, AND, OR" from left to right

Error Handling
If search clause's parameters number is not correct, field name is not existing, or there is not supported operator, it generates SearchException.

example(s):

```
SELECT * FROM students
WHERE PKID > 4
```

```
SELECT * FROM STUDENTS INNER JOIN INSTRUCTOR
WHERE PKID > 2 && first_name LIKE e
```

ORDERBY Command

(some select command)
(some where command (optional))
ORDERBY column direction

- The user is limited to only one field/column to order by
- The values for direction is limited to ASC for ascending and DESC for descending
- The command can be written both “ORDER BY” or “ORDERBY”

example:

```
SELECT * FROM students
ORDERBY names ASC
```

UPDATE Command

- UPDATE primaryKey IN tablename SET column TO value
- The user can only update a single field in a table at a time
- The user must specify the primary key of the row they wish to update

example:

```
SELECT * FROM students
ORDERBY names ASC
```

INSERT Command

```
INSERT tablename
(
PKcolumn1 value,
column2 value,
FKcolumn3 value,
...
);
```

- The user must enter values for all columns that exist in the table

example:

```
INSERT students
(
PKID 13,
names Frank,
FKID 4,
);
```

DELETE Commands

```
DELETE ALL ROWS tablename
DELETE TABLE tablename
DELETE primaryKey FROM tablename
```

example(s):

```
DELETE ALL ROWS students
```

```
DELETE TABLE students
```

```
DELETE 13 FROM students
```

SECTION C: Test Cases

ID	1
Title	Create a Table and Columns
Pre-conditions	1. The user should think about what columns will be in the table and which ONE, AND ONLY ONE, column will be the primary key 2. The primary key column should be type int and the first column to be created 3. Foreign key, unlike primary key, isn't mandatory 4. The only data types allowed are int, double, String, and Date
Test Steps	1. Type "**CREATE TABLE *tablename* (**" in the text area 2. Type "***columnname datatype,***" 3. The first columnname must start with the letters "**PK**" (PKcolumnname) 4. To add more columns just repeat step 2 for each new column 5. No comma is necessary after the last column, instead of a comma, type "**);**" 6. Click on the Run button
Expected Results	1. A text file with the name tablename.txt should appear inside the dbfile folder 2. The column name(s), column data type(s), indication whether column is primary key, foreign key, or normal (neither pk or fk), and field value(s) should be stored in the text file
Post-conditions	1. Type "**DELETE TABLE *tablename***" in the text area to delete a table from database 2. Table structure can't be changed after creation
Test Data	N/A
Priority (Low, Medium, High)	High
Author	Joy Cao

ID	2
Title	Insert a Record
Pre-conditions	1. The values being inserted into each column should match the data type declared for that column 2. Primary key names must start with the letters "PK" 3. Foreign key names must start with the letters "FK"
Test Steps	1. Type "**INSERT *testTable* (**" in the text area to insert a record into the table testTable 2. Type "***pkid 1,***" to insert the value "1" into the column "pkid" 3. Type "***test one***" to insert the value "one" into the column "test" 4. type "**);**" 5. Click the Run button
Expected Results	1. The text file "testTable.txt" should now contain one record with values "1 one"
Post-conditions	1. Type "**DELETE *pkid* FROM *testTable***" to delete a record from the table 2. Type "**DELETE ALL ROWS *testTable***" to empty testTable of its records but retain its structure

Test Data	testTable.txt
Priority (Low, Medium, High)	High
Author	Joy Cao

ID	**3**
Title	Update Field
Pre-conditions	1. User should know the primary key value of the record which the field to be updated belongs to 2. User should also know the column name and table name of the field to be updated
Test Steps	1. Type "**UPDATE *1* IN *testTable* SET *test* TO *two***" to update the field under the column "test" and belonging to the record whose primary key (pkid) is "1" to the value "two" 2. Click the Run button
Expected Results	1. Inside text file "testTable.txt", the first record should now be "1 two" 2. This change should be observed in the display area as well
Post-conditions	1. To change the value back, the user should know what the original field value was and follow the steps to update field
Test Data	testTable.txt
Priority (Low, Medium, High)	Medium
Author	Joy Cao

ID	**4**
Title	Display Table
Pre-conditions	N/A
Test Steps	1. To show only columns "names" and "first_name", type "**SELECT *names, first_name* FROM *students* INNER JOIN *instructor***" 2. To show only records where pkid is smaller than "6", type "**WHERE *pkid* < 6**" 3. To show records where names contain the letter "o", type "**AND *names* LIKE *o***" 4. To display the results in ascending order based on names, type "**ORDERBY *names* ASC**", without the ASC is also fine since ascending order is the default 5. Click the Run button
Expected Results	1. The display area should contain a table with two columns (names and first_name) and two records 2. The first record should be "Joy Mehwish" 3. Second record should be "Ronnie Stephen"
Post-conditions	N/A
Test Data	students.txt instructor.txt
Priority (Low, Medium, High)	Low
Author	Joy Cao