

CSCE 310: Database Systems HW 2

Raymond Zhu

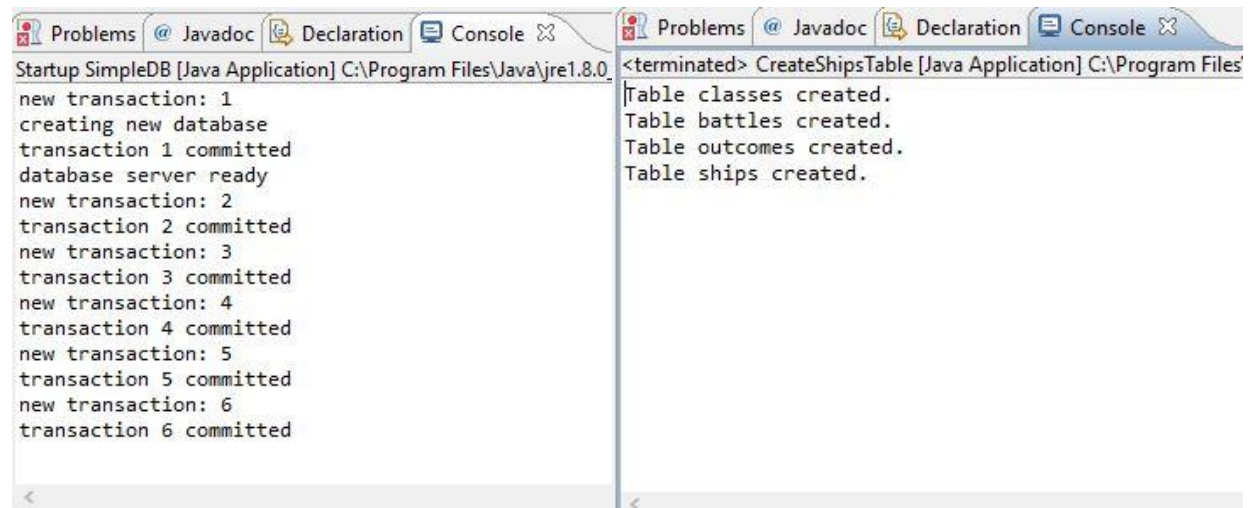
923009555

1.

```
{ acctNo = '12345' | type = 'savings', balance = 12000; }  
{ acctNo = '23456' | type = 'checking', balance = 1000; }  
{ acctNo = '34567' | type = 'savings', balance = 25; }
```

2.

```
public class CreateShipsTable {  
    public static void main(String[] args) {  
        Connection conn = null;  
        try {  
            Driver d = new SimpleDriver();  
            conn = d.connect("jdbc:simpledb://localhost", null);  
            Statement stmt = conn.createStatement();  
  
            String s = "CREATE TABLE classes (class varchar(20),type varchar(2),country varchar(20),numGuns int,bore int,disp int);"  
            stmt.executeUpdate(s);  
            System.out.println("Table classes created.");  
  
            s = "CREATE TABLE battles (name varchar(20), date varchar(20))";  
            stmt.executeUpdate(s);  
            System.out.println("Table battles created.");  
  
            s = "CREATE TABLE outcomes (ship varchar(20),battle varchar(20),result varchar(10))";  
            stmt.executeUpdate(s);  
            System.out.println("Table outcomes created.");  
  
            s = "CREATE TABLE ships (name varchar(20), class varchar(20), launched int);"  
            stmt.executeUpdate(s);  
            System.out.println("Table ships created.");  
        }  
        catch(SQLException e) {  
            e.printStackTrace();  
        }  
        finally {  
            try {  
                if (conn != null)  
                    conn.close();  
            }  
            catch (SQLException e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}
```



3.

select SId, Sname from student;

- The command is selecting the id and name of every student in the student table. The students are ordered by SId in the projection.

```
SQL> select SId, Sname from student;
```

sname	sid
joe	1
amy	2
max	3
sue	4
bob	5
kim	6
art	7
pat	8
lee	9

```
SQL> |
```

select EId from section, enroll;

- The enroll table is scanned in the inner loop. For every section, the command will grab every enrollment id in the enroll table.

```
SQL> select EId from section, enroll;
```

eid
14
24
34
44
54
64
14
24
34
44
54
64
14
24
34
44
54
64
14
24
34
44
54
64
14
24
34
44
54
64

```
SQL>
```

Select EId from enroll, section

- Since the section table is in the inner loop, for each enrollment id, the command will return the same enrollment id x number times – where x is the number of sections in the section table.

```
SQL> select EId from enroll, section;
```

```
|
  eid
-----
  14
  14
  14
  14
  14
  24
  24
  24
  24
  24
  34
  34
  34
  34
  34
  44
  44
  44
  44
  44
  54
  54
  54
  54
  54
  64
  64
  64
  64
  64
```

```
SQL>
```

select Title, SectId from course, section where CId=CourseId and Prof='turing';

- Since both tables share a relation given the course id, the command returns the title and section id for every course id that has professor turing.

```
SQL> select Title, SectId from course, section where CId=CourseId and Prof='turing';
```

```
sectid      title
-----
   13      db systems
   23      db systems
```

```
SQL>
```

select SName,GradYear from student, dept where MajorId = DId and DId=20;

- The command is selecting the student name and graduation year attribute from the student table where each student 's department id is equal to 20. A relation is shared between MajorId and DId for both tables.

SQL> select SName,GradYear from student, dept where MajorId = DId and DId=20;

gradyear	sname
2004	amy
2005	sue
2001	kim
2001	pat

SQL>

select SName,GradYear from student, dept where MajorId = 20;

- Not the same as the previous. The command is selecting the SName and GradYear of each student in the student table where the student's MajorId is equal to 20. Since dept is nested in the inquiry, and no relation is specified between the two tables, each student is repeated by the number of departments in the dept table.

SQL> select SName,GradYear from student, dept where MajorId = 20;

gradyear	sname
2004	amy
2004	amy
2004	amy
2005	sue
2005	sue
2005	sue
2001	kim
2001	kim
2001	kim
2001	pat
2001	pat
2001	pat

SQL>

4a.

$\pi_{class, country}(\sigma_{bore \geq 16}(Classes))$

Class	Country
IOWA	USA
NORTH CAROLINA	USA
YAMATO	JAPAN

4b.

$\pi_{name}(\sigma_{launched < 1921}(Ships))$

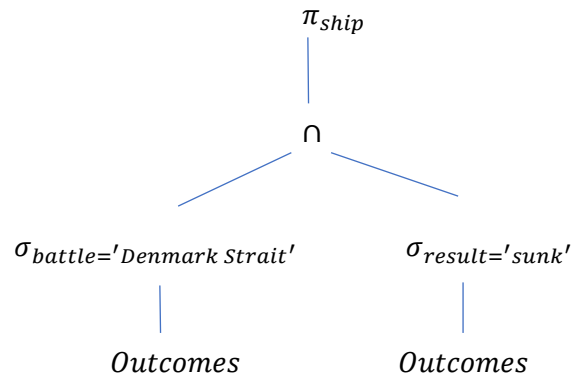
Name
Haruna
Hiei
Kirishima

Kongo
Ramillies
Renown
Repulse
Resolution
Revenge
Royal Oak
Royal Sovereign
Tennessee

4c.

$\pi_{ship}(\sigma_{battle='Denmark Strait' \text{ AND } \sigma_{result='sunk'}(Outcomes))$

Ship
Bismarck
Hood

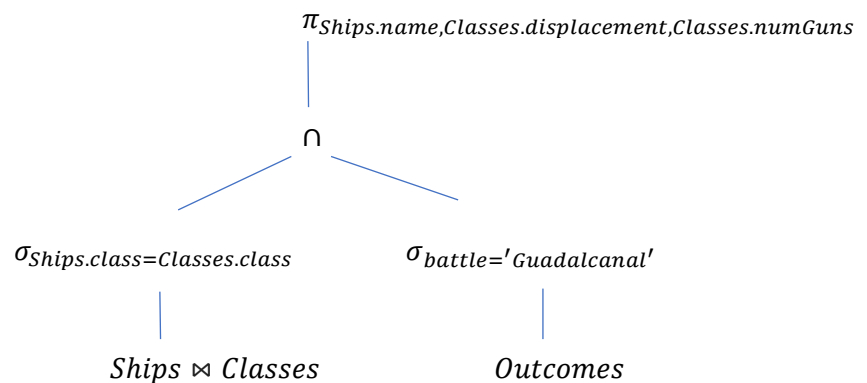


4e.

S = Ships & C = Classes

$\pi_{S.name, C.displacement, C.numGuns}(\sigma_{S.class=C.class}(S \bowtie C) \cap \sigma_{battle='Guadalcanal'}(Outcomes))$

Name	Displacement	Number of Guns
Kirishima	32000	8
Washington	37000	9



5a.

$$\pi_{class}(Classes) \subseteq \pi_{class}(\sigma_{bore \leq 16}(Classes))$$

5b.

$$\pi_{class}(\sigma_{numGuns > 9}(Classes)) \subseteq \pi_{class}(\sigma_{bore \leq 14}(Classes))$$

6.

```
C:\Apache\db-derby-10.13.1.1-bin\bin>java -version
java version "1.8.0_144"
Java(TM) SE Runtime Environment (build 1.8.0_144-b01)
Java HotSpot(TM) 64-Bit Server VM (build 25.144-b01, mixed mode)
```

7.

```
C:\Apache\db-derby-10.13.1.1-bin\bin>java org.apache.derby.tools.sysinfo
----- Java Information -----
Java Version: 1.8.0_144
Java Vendor: Oracle Corporation
Java home: C:\Program Files\Java\jre1.8.0_144
Java classpath: C:\Apache\DB-DER~1.1-B\lib\derby.jar;C:\Apache\DB-DER~1.1-B\lib\derbytools.jar;C:\Apache\DB-DER~1.1-B\lib\derbyoptionaltools.jar;C:\Apache\db-derby-10.13.1.1-bin\lib\derby.jar;C:\Apache\db-derby-10.13.1.1-bin\lib\derbytools.jar;.
OS name: Windows 10
OS architecture: amd64
OS version: 10.0
Java user name: rayzr
Java user home: C:\Users\rayzr
Java user dir: C:\Apache\db-derby-10.13.1.1-bin\bin
java.specification.name: Java Platform API Specification
java.specification.version: 1.8
java.runtime.version: 1.8.0_144-b01
----- Derby Information -----
[C:\Apache\db-derby-10.13.1.1-bin\lib\derby.jar] 10.13.1.1 - (1765088)
[C:\Apache\db-derby-10.13.1.1-bin\lib\derbytools.jar] 10.13.1.1 - (1765088)
[C:\Apache\db-derby-10.13.1.1-bin\lib\derbyoptionaltools.jar] 10.13.1.1 - (1765088)
-----
----- Locale Information -----
Current Locale : [English/United States [en_US]]
Found support for locale: [cs]
version: 10.13.1.1 - (1765088)
Found support for locale: [de_DE]
version: 10.13.1.1 - (1765088)
Found support for locale: [es]
version: 10.13.1.1 - (1765088)
```

[illegible]

9a.

```
ij> Select MODEL From PC Where SPEED>=3.00;
```

```
MOD&I
```

```
----
```

```
1005
```

```
1006
```

```
1013
```

9b.

```
ij> Select MAKER From PRODUCT, LAPTOP Where PRODUCT.MODEL=LAPTOP.MODEL AND HD>=100;
```

```
MAK&
```

```
----
```

```
A
```

```
B
```

```
E
```

```
F
```

```
G
```

9c.

```
ij> Select PC.MODEL,PC.PRICE From PRODUCT,PC Where PRODUCT.MAKER='B' and  
PRODUCT.MODEL=PC.MODEL Union Select LAPTOP.MODEL,LAPTOP.PRICE From  
PRODUCT,LAPTOP Where PRODUCT.MAKER='B' and PRODUCT.MODEL=LAPTOP.MODEL  
Union Select PRINTER.MODEL,PRINTER.PRICE From PRODUCT,PRINTER Where  
PRODUCT.MAKER='B' and PRODUCT.MODEL=PRINTER.MODEL;
```

```
MOD&IPRICE
```

```
-----
```

```
1004|649
```

```
1005|630
```

```
1006|1049
```

```
2007|1429
```


10.

```
SQL> run ships.sql
```

```
ships.sql
```

```
1 records processed
```

```
1 records processed
```

```
1 records processed
```

```
1 records processed
```

```
1 records processed
```

```
1 records processed
```

```
1 records processed
```

```
1 records processed
```

```
1 records processed
```

```
1 records processed
```

```
1 records processed
```

```
1 records processed
```

```
else if (cmd.startsWith("run"))
```

```
{
```

```
    String[] list = cmd.split(" ");
```

```
    String shit = "";
```

```
    System.out.println(list[1]);
```

```
    BufferedReader brr = new BufferedReader(new FileReader(list[1]));
```

```
    try {
```

```
        StringBuilder stringBuilder = new StringBuilder();
```

```
        String line = brr.readLine();
```

```
        while (line != null) {
```

```
            stringBuilder.append(line);
```

```
            line = brr.readLine();
```

```
        }
```

```
        shit = stringBuilder.toString();
```

```
    } finally {
```

```
        brr.close();
```

```
    }
```

```
    String[] list2 = shit.split(";");
```

```
    for(String s : list2)
```

```
    {
```

```
        doUpdate(s);
```

```
    }
```

```
}
```

An aggie does not lie, cheat, steal, or tolerate those who do.