

## INFS1200/7900 Mock QUIZ 2

Name: \_\_\_\_SAMPLE SOLUTION\_\_\_\_ Student #: \_\_\_\_\_

Signature: \_\_\_\_\_

### Notes about this examination

1. You have **90 minutes** (1 hour and 30 minutes) to write this examination.
2. Write your name, student #, and signature in ink (pen). You may use a pencil to write your solutions.
3. Answer all the questions on this paper.
4. The marks for each question are given in [].
5. Good luck!

Questio	Mark	Max
Q1		10
Q2		15
Q3		20
Q4		15
Q5		15
Total		75

### Functional Dependencies and Normal Forms

Q1. [10 marks] Answer the following questions.

Q1A. Suppose you are given a relation  $R(A,B,C,D)$  with the following functional dependencies:

$AB \rightarrow C$ ,  $AB \rightarrow D$ ,  $C \rightarrow A$ ,  $D \rightarrow B$

Is  $R$  in BCNF? If not, decompose this relation into BCNF using the algorithm we covered in class and in the book; circle all answers in your final decomposition. Show all your work (5 marks).

Candidate Keys:  $\{A,B\}$ ,  $\{A,D\}$ ,  $\{B,C\}$ ,  $\{C,D\}$

Final Relations:

$R_1 [C, A]$

FD1:  $C \rightarrow A$

$R_2 [D, B]$

FD1:  $D \rightarrow B$

$R_3 [C, D]$

No Non-trivial FD

Q1B. Suppose you are given a relation  $R(A,B,C,D)$  with the following functional dependencies:

$AB \rightarrow C$ ,  $AB \rightarrow D$ ,  $C \rightarrow A$ ,  $D \rightarrow B$

Is  $R$  in 3NF? If not, decompose this relation into 3NF using the algorithm we covered in class and in the book; circle all answers in your final decomposition. Show all your work (5 marks).

Candidate Keys:  $\{A,B\}$ ,  $\{A,D\}$ ,  $\{B,C\}$ ,  $\{C,D\}$

The relation  $R$  is already in 3NF.

The rest of the questions on this quiz are related to the database schema and database instance, which is on Appendix A. Please review it carefully and answer all of the following questions. For each query remove duplicates from your final answers where they are not explicitly requested, and include no extra columns).

2. [15 marks] Write the following SQL queries **without using subqueries**.

2A. Find the name of the ships that have more than 8 guns (4 marks).

Query
<pre>SELECT S.shipName FROM Ships S, ShipModels M WHERE S.model = M.model AND M.numGuns &gt; 8;</pre>

2B. Find those countries that have ship models with both less than 7 and greater than 7 guns. (Assuming the intersect operator is **not** implemented). Show the result of your query using data from Appendix A (5 marks).

Query	Result
<pre>SELECT DISTINCT M1.country FROM ShipModels M1, ShipModels M2 WHERE M1.country = M2.country AND M1.numGuns &lt; 7 AND M2.numGuns &gt; 7;</pre>	<p>Britain</p>

2C. Find the number of ships that participated in each battle. Show the result of your query using data from Appendix A (6 marks).

Query	Result
<pre>SELECT battleName, COUNT(*) FROM Outcomes GROUP BY battleName;</pre>	<p>Surigao Strain, 3 Guadalacanal, 2 North Cape, 2</p>

3. [20 marks] Write the following SQL queries using **at least one subquery**. Show the result of your query using data from Appendix A for parts 2A and 2B.

3A. Find the models of ships, at least one of which was ‘sunk’ in a battle (6 marks).

Query	Result
<pre>SELECT DISTINCT S.model FROM Ships WHERE shipName IN   (SELECT shipName    FROM Outcomes    WHERE result = "sunk");</pre>	<p>Kongo North Carolina Revenge</p>

3B. Find the name of the heaviest ship model(s) (7 marks).

Query	Result
<pre>SELECT model FROM ShipModels WHERE displacement &gt;= ALL   (SELECT displacement    FROM ShipModels);</pre> <p>DO NOT USE ORDER BY and LIMIT 1.</p>	<p>Yamato</p>

3C. Find countries with at least two ship models, where at least one of the models has more guns than the average number of guns across all of the ship models (7 marks).

Query
<pre>SELECT DISTINCT country FROM ShipModels GROUP BY country HAVING COUNT(*) &gt; 1 AND MAX(numGuns) &gt;   (SELECT AVG(numGuns)    FROM ShipModels);</pre>

4. [15 marks] Write the following SQL queries. You can use any of the operators taught in the lectures.

4A. Find the name of ships for which there exists some other ship with the same model that was launched after 1943. For example Missouri would be on the list since Wisconsin is of the same model (Iowa) and was launched after 1943 (7 marks).

Query

```
SELECT S1.shipName
FROM Ships S1, Ships S2
WHERE S1.model = S2.model
AND S1.shipName <> S2.shipName
AND S2.launched > 1943);
```

4B. Find the models of ships that exactly two of which fought in the same battle or have at least 3 different ships. Show the result of your query using data from Appendix A (8 marks).

Query

```
SELECT S.model
FROM Ships S
WHERE S.model IN
    (SELECT model
     FROM Outcomes O, Ships S
     WHERE O.shipName = S.shipName
     GROUP BY O.battleName, S.model
     HAVING COUNT(*) = 2)
OR S.model IN
    (SELECT model
     FROM Ships
     GROUP BY model
     HAVING COUNT(*) > 2);
```

Result

Tennessee  
Kongo  
Iowa  
Revenge

5. [15 marks] The following questions are related to modifying data and integrity constraints.

5A. Delete from table Ships all ships sunk in battles (3 marks).

Query

```
DELETE FROM Ships
WHERE shipName IN
(SELECT shipName
FROM Battles
WHERE result = "sunk");
```

5B. Modify the shipModels relation so that gun bores are measured in centimeters instead of inches. Note that the attribute type of bore is double. (one inch = 2.5 centimeters) (3 marks)

Query

```
ALTER TABLE ShipModels
ALTER COLUMN bore DOUBLE(3, 1);
UPDATE ShipModels
SET bore = bore * 2.5;
```

Assuming bore can be at most three digits before decimal point and one digit after the decimal point.

5C. Write a query for creating the Ships table that would enforce the launched year to be between 1000 and 2020. You may assume any reasonable format/domain for the fields (4 marks).

Query

```
CREATE TABLE Ships (
    shipName VARCHAR(1000) NOT NULL,
    model VARCHAR(1000),
    launched INT,
    PRIMARY KEY(shipName),
    FOREIGN KEY (model) REFERENCES ShipModels(model),
    CONSTRAINT launchedCheck CHECK (launched >= 1000 AND launched < 2020));
```

5D. Add a constraint that no more than 3 ships of the same model can participate in a single battle (5 marks).

Query

```
CREATE ASSERTION battleModelCap as CHECK
(NOT EXISTS
    (SELECT *
    FROM Ships S, Outcomes O
    WHERE S.shipName = O.shipName
    GROUP BY O.battleName, S.model
    HAVING COUNT(*) > 3));
```