CS 4347.004
Daniel Crawford

6.1)
Tables/Relations in SQL are different because in SQL, there can be multiple tuples that are completely identical in their attribute values(duplicate tuples). The reason SQL allows duplicate tuples is because duplicate elimination can be an expensive operation. Another reason is that users may want to see duplicate tuples when they query the database.

6.3)
Entity Integrity and referential integrity can be specified within a database using the CREATE TABLE statement after the attributes are declared. This is also acceptable for the ALTER TABLE command. SQL can set a key as a PRIMARY KEY and a FOREIGN KEY after they are declared to accomplish these constraints. For referential triggered actions, SQL has several commands to change the action when it is called. These commands to react to are ON DELETE and ON UPDATE, and they can react by using SET NULL, CASCADE and SET DEFAULT.

6.12)
a)
```
SELECT      Name
FROM        STUDENT
WHERE       Major=’CS’ AND Class=4
```
b)
```
SELECT      COURSE.Course_Name
FROM        SECTION, COURSE
WHERE       SECTION.Instructor=’King’ AND
            (SECTION.Year=’07’ OR SECTION.Year = ‘08’)
```
c)
```
SELECT      Course_number, Year, Semester, COUNT(*)
FROM        SECTION, GRADE_REPORT
WHERE       Instructor=’King’ AND
                    SECTION.Section_identifier=GRADE_REPORT.Section_identifier
GROUP BY    Course_number, Semester, Year
```
d)
```
SELECT      Name, COURSE.Course_Name, COURSE.Course_Number, COURSE.Credit_hours,
            SECTION.Semester, SECTION.Year, GRADE_REPORT.Grade
FROM        STUDENT, COURSE, SECTION, GRADE_REPORT
WHERE       STUDENT.Class = 4 AND STUDENT.Major=’CS’ AND
            STUDENT.Student_number = GRADE_REPORT.Student_number AND
            GRADE_REPORT.Section_identifier = SECTION.Section_Identifier AND
            SECTION.Course_number = COURSE.Course_number
```

7.1)
There are six clauses in an SQL retrieval query but only 2, SELECT and FROM, are mandatory. The optional ones are WHERE, GROUP BY, HAVING, ORDER.

- SELECT: Lists the attributes or functions to be retrieved.
- FROM: Specifies all relations/tables needed in the query
- WHERE: Specifies conditions for selecting some of the tuples
- GROUP BY: Specifies grouping attributes

- HAVING: Specifies a condition on the groups being selected rather than on the individual tuples
- ORDER BY: Specifies an order for displaying the result of a query

7.3)
SQL considers each NULL value to be distinct from each other NULL value. Therefore, instead of using comparison operators such as == or <> it uses IS and IS NOT. Since each NULL is distinct, a comparison operator is not necessary in this situation. In an aggregate function, NULL values are discarded when they are applied. Grouping on the other hand will group all the NULL values into their own group, for all values that have a NULL for any attribute.

7.5)
a)

```
SELECT      Dname, COUNT(*)
FROM        EMPLOYEE, DEPARTMENT
WHERE       Dnumber=Dno
HAVING      Salary > 30000
```

Query Result:
Dname = Research, Count = 2
Dname = Administration, Count = 1
Dname = Headquarters, Count = 1

b)
Can be done using a nested clause

```
SELECT      Dname, COUNT(*)
FROM        DEPARTMENT, EMPLOYEE
WHERE       Dnumber = Dno AND Sex = M AND Dno IN (
            SELECT      Dno
            FROM        EMPLOYEE
            WHERE Salary > 30000)
GROUP BY    Dname
```

Query Result:
Dname = Research, Count = 2
Dname = Administration, Count = 0
Dname = Headquarters, Count = 1

8.7)
An OUTER JOIN has multiple ways it can be joined, such as the LEFT OUTER JOIN and the RIGHT OUTER JOIN. For these, every tuple in the left/right table must appear in result, and if there is no matching tuple, pad with NULL values for attributes of left table. An INNER JOIN will include a tuple in the result only if a matching tuple exists in the other relation.

8.9)
Relational calculus in general specifies what needs to be retrieved rather than how to retrieve it. Therefore Domain and Tuple relational calculus are very similar. However there is one big difference

between the two. In Domain Relational Calculus, rather than having variables range over tuples, variables range over single values from domains of attributes.

8.18)
a)How many copies of the book titled The Lost Tribe are owned by the library branch whose name is ‘Sharpstown’?

TEMP ← BOOK * BOOK_COPIES * LIBRARY_BRANCH
RESULT ← pi_(No_of_copies)((sigma_(Title=’The Lost Tribe’ AND branch_name=’Sharpstown’)(TEMP))

b) How many copies of the book titled The Lost Tribe are owned by each library branch

NUMBER_OF_BOOKS ← sigma_(Title=’The Lost Tribe’)(BOOK)
RESULT ← pi_(Branch_id, No_of_copies)(NUMBER_OF_BOOKS * BOOK_COPIES)

c) Retrieve the names of all borrowers who do not have any books checked out.

NO_CHECKOUT ← sigma_(Card_no)(BORROWER) – sigma_(Card_no)(BOOK_LOANS)
RESULT ← pi_(Name)(NO_CHECKOUT * BORROWER)

d) For each book that is loaned out from the Sharpstown branch and whose Due_date is today, retrieve the book title, the borrower’s name, and the borrower’s address.

SHARPSTOWN ← pi_(Branch_id)(sigma_(Branch_name=’Sharpstown’)(LIBRARY_BRANCH))
DUE_TODAY ← sigma_(Due_date=’Today’)(BOOK_LOANS)
SHARPSTOWN_BOOKS ← sigma_(Branch_id, Card_no)(SHARPSTOWN * DUE_TODAY)
RESULT ← sigma(Title,Name,Address)(SHARPSTOWN_BOOKS * BOOKS * BORROWER)

e) For each library branch, retrieve the branch name and the total number of books loaned out from that branch

LIBRARY_LOANED ← BranchID ℑ Count_(Book_ID)(BOOK_LOANS)
RESULT ← pi_(Branch_name, Count_Bood_ID)(LIBRARY BRANCH * LIBRARY_LOANED)

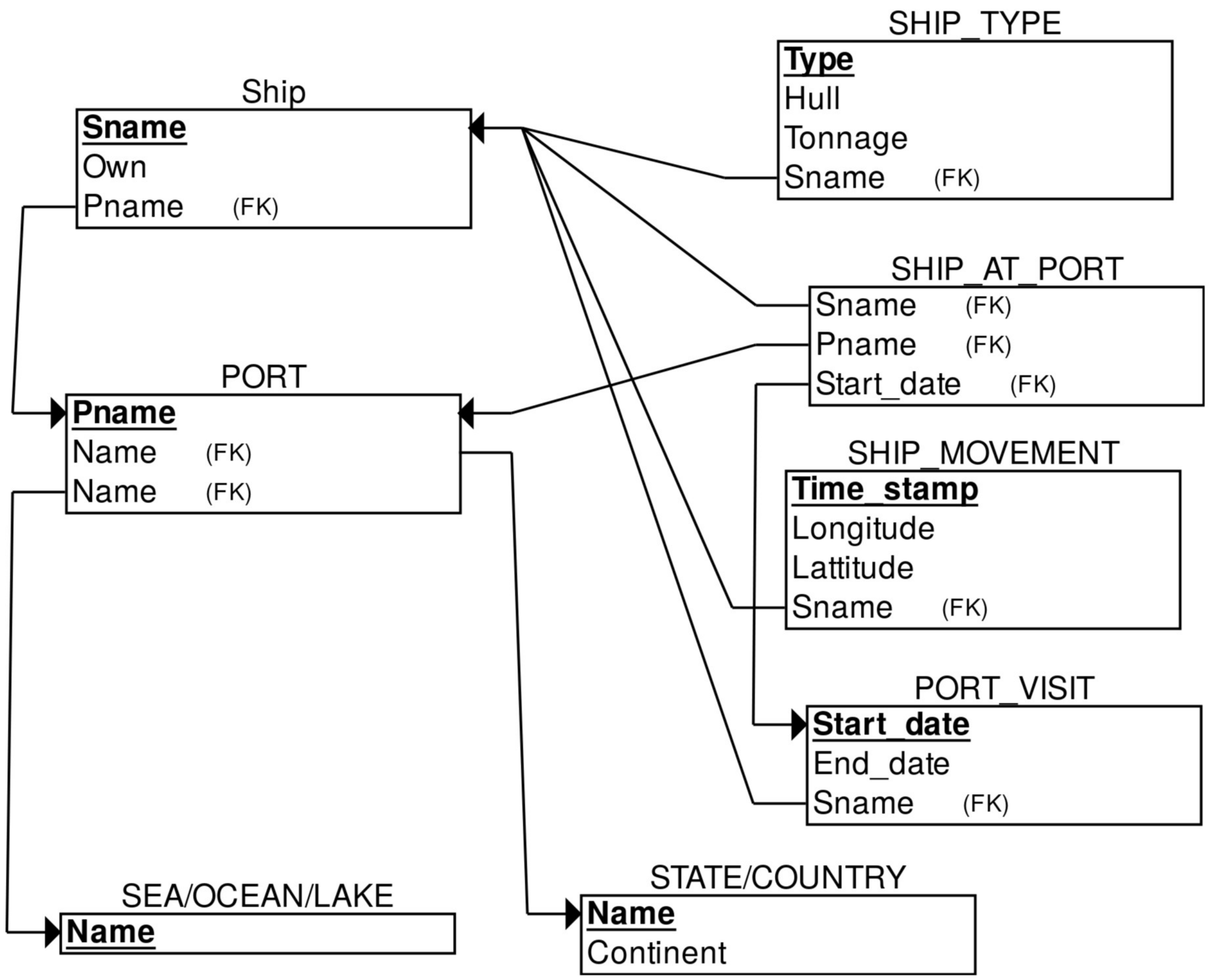
f) Retrieve the names, addresses, and number of books checked out for all borrowers who have more than five books checked out

BOOKS_BORROWED ← Card_no ℑ Count_(Book_id)(BOOK_LOANS)
FIVE_BOOKS_B ← sigma_(Count_Book_id)(BOOKS_BORROWED)
RESULT ← pi_(Name,Address,Count_Book_id)(FIVE_BOOKS_B * BORROWER)

g) For each book authored (or coauthored) by Stephen King, retrieve the title and the number of copies owned by the library branch whose name is Central

STEPHEN_KING = sigma_(Author_name=’Stephen King’)(BOOK)
CENTRAL = pi_(Branch_name=’Central’)(LIBRARY_BRANCH)
STEPHEN_KING_BOOKS = sigma_(Book_id, Title)(BOOK * STEPHEN_KING)
RESULT ← pi_(Title, No_of_copies)(STEPHEN_KING_BOOKS * CENTRAL * BOOK_COPIES)

9.4)


SHIP_TYPE
Type
Hull
Tonnage
Sname (FK)
Ship
Sname
Own
Pname (FK)
SHIP_AT_PORT
Sname (FK)
Pname (FK)
Start_date (FK)
PORT
Pname
Name (FK)
Name (FK)
SHIP_MOVEMENT
Time_stamp
Longitude
Lattitude
Sname (FK)
PORT_VISIT
Start_date
End_date
Sname (FK)
SEA/OCEAN/LAKE
Name
STATE/COUNTRY
Name
Continent