**ISS4014 – Database Systems and Web Integration**

**Chapter 03 – Activities and Homework**

|  |  |
| --- | --- |
| **Name:** | Logan Strong |
| **Date:** | January 30th, 2024 |

**Chapter 03 REVIEW (5 points)**

Respond to the following Chapter 03 review questions (you may have discussed some of these questions in class, so be sure to include ideas and information you gained from the class discussion).

1. (1pt) What does it mean to say that a database displays entity and referential integrity? *(The best definition is in the side column of the text, pages 74 and 75.)*

|  |  |  |  |
| --- | --- | --- | --- |
| **Term** | **Page #s**  **In Book** | **Using Book Language** | **In your own words** |
| Entity integrity | #74 | The property of a relational table that guarantees each entity has a unique value in a primary key and that the key has no null values | It’s a classification for if every entity using a primary key in a table points to values that aren’t empty |
| Referential Integrity | #75 | A condition by which a dependent table’s foreign key must have either a null entry or a matching entry in the related table | A condition where one tables foreign key doesn’t point to an entry in another table that doesn’t exist. |

1. (1pt) What are the **requirements** that **two relations** must satisfy to be considered **union-compatible**?

|  |
| --- |
| That they have the same number of columns and the corresponding columns have compatible domains. |

1. (1pt) Which relational algebra **operators** can be applied to a **pair of tables** that are **not** union-compatible?

|  |
| --- |
| Product, Join, and Divide |

*For questions 4 & 5 below, load the* ***CH03\_CollegeQue*** *database using SQL Workbench.*

1. (1pt) Use the following MySQL query to create the table resulting from **πstu\_code(student)**.

**Select Stu\_Code**

**from Student;**

Copy and paste a screenshot of the query result below:

|  |
| --- |
| ***<Paste Screenshot Here>*** |

A screenshot of a computer

Description automatically generated

1. (1pt) Create the table that would result from **πdept\_code, stu\_code(student ⋈ professor)**.

Use the following MySQL query to generate this table:

**Select Dept\_Code, Stu\_Code**

**from Student, Professor**

**where Student.Prof\_Code = Professor.Prof\_Code;**

Copy and paste a screenshot of the query result below:

|  |
| --- |
| ***<Paste Screenshot Here>*** |

A screenshot of a computer

Description automatically generated

**Chapter 03 PROBLEMS (25 points)**

Use the database shown in the figure below to answer problems 1-5.

|  |  |
| --- | --- |
| **Table** | **Data** |
| TRUCK |  |
| BASE |  |
| TYPE |  |

1. (2 pts) For each table, identify the primary key and the foreign key(s). Write N/A (not applicable) if a table does not have a foreign key.

|  |  |  |  |
| --- | --- | --- | --- |
| Table | Primary Key | Foreign Keys | Table that each Foreign Key Relates to: |
| TRUCK | TRUCK\_NUM | BASE\_CODE, TYPE\_CODE | BASE, TYPE |
| BASE | BASE\_CODE | NA | NA |
| TYPE | TYPE\_CODE | NA | NA |

1. (2 pts) Do the tables exhibit entity integrity? Answer yes or no and then explain your answer.

|  |  |  |
| --- | --- | --- |
| Table | Entity Integrity?  (Yes/No) | Explanation |
| TRUCK | NO | Has a null value relating to base code for truck\_num 1004 |
| BASE | NA | NA |
| TYPE | NA | NA |

1. (2 pts) Do the tables exhibit referential integrity? Answer yes or no, and then explain your answer. *Write N/A (Not Applicable) if the table does not have a foreign key.*

|  |  |  |
| --- | --- | --- |
| Table | Referential Integrity?  (Yes/No) | Explanation |
| TRUCK | YES | Doesn’t reference a value that doesn’t exist in the related tables (501, 502, 503 all exist in the base table and same for the values in type\_code) |
| BASE | NA | NA |
| TYPE | NA | NA |

1. (2 pts) What are the type(s) of relationship(s) between TRUCK, BASE, and TYPE? (How are they related in terms of business rules and cardinality?)

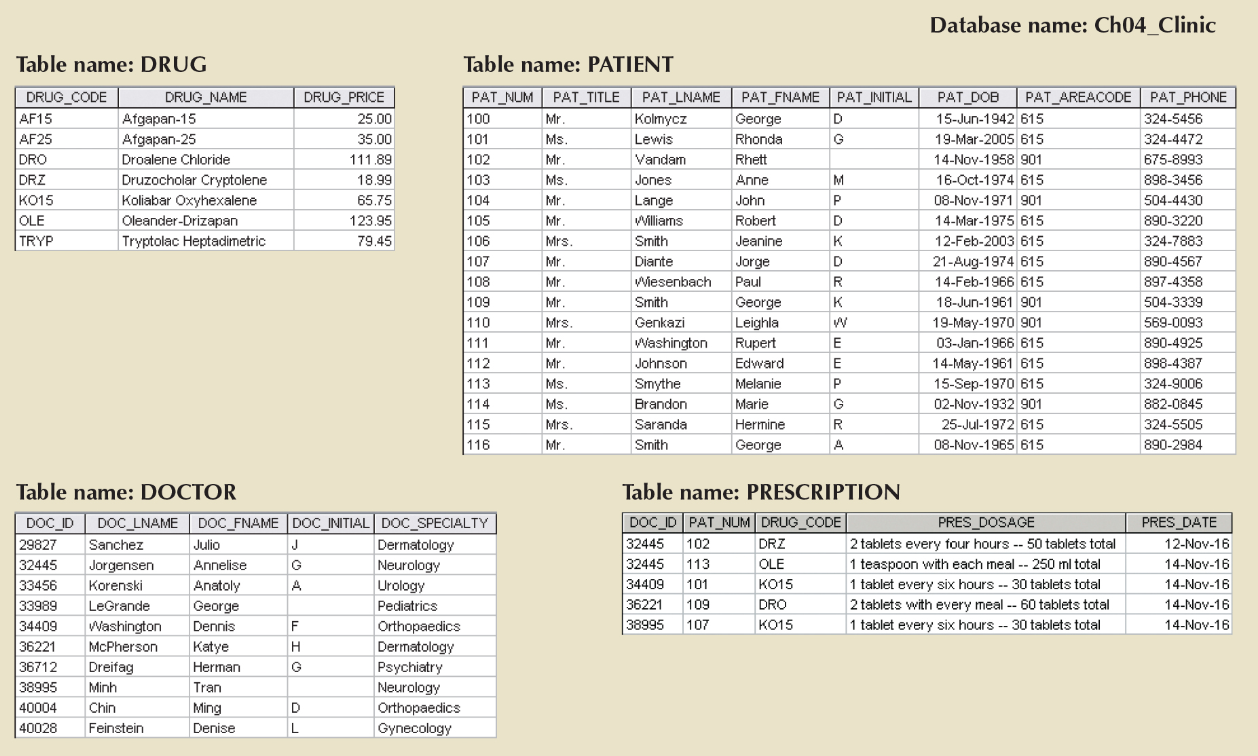
|  |
| --- |
| They are all One to Many. Many trucks can have one base type, but a truck can only have one base. Many trucks can be the same type, but each truck can only be assigned one type. |

1. (2.5 pts) Using Visio, create a crow’s foot ERD to show the relationship between TRUCK, BASE, and TYPE\_CODE. Include the attributes and indicate the cardinality of the relationships. Copy and paste a screenshot of the ERD below:

|  |
| --- |
| ***<Paste Screenshot Here>*** |



Use the database shown in the figure below to answer Problems 6-12. (The database can be created using the Ch04\_Clinic\_MySQL script in the student files. Some dates may differ from the image).



1. (2 pts) For each table, where possible, identify the most appropriate **primary** key:

|  |  |
| --- | --- |
| Table | Primary Key |
| DRUG | DRUG\_CODE |
| PATIENT | PAT\_NUM |
| DOCTOR | DOC\_ID |
| PRESCRIPTION | NONE |

1. (2 pts) For each table, where possible, identify a **candidate** key (not using the primary key identified above). (List of columns or combination of columns that practically could have been used as a primary key, usually 2 columns or 3. If the combination of columns becomes very long or almost includes all the columns, that is a hint that it may not be a practical candidate key. In this list you create, do not list the primary key as we want to identify other possible candidate keys.)

|  |  |
| --- | --- |
| Table | Possible Candidate Key |
| DRUG | DRUG\_NAME |
| PATIENT | PAT\_TITLE, PAT\_LNAME, PAT\_FNAME |
| DOCTOR | DOC\_LNAME, DOC\_FNAME |
| PRESCRIPTION | DOC\_ID, PAT\_NUM, DRUG\_CODE |

1. (2 pts) For each table, where possible, identify the **foreign** key(s):

|  |  |  |
| --- | --- | --- |
| Table | Foreign Key(s) | Table that each Foreign Key Relates to: |
| DRUG | NA | NA |
| PATIENT | NA | NA |
| DOCTOR | NA | NA |
| PRESCRIPTION | DOC\_ID, PAT\_NUM, DRUG\_CODE | DOCTOR, PATIENT, DRUG |

1. (2.5 pts) Using Visio or Workbench, create a crow’s foot ERD. Include the attributes and indicate the cardinality of the relationships. Paste an image of the ERD below.

|  |
| --- |
| ***<Paste Screenshot Here>*** |



1. (2 pts) Create the table that would result from applying the SELECT and PROJECT relational operators to the DRUG table to return only the DRUG\_CODE and DRUG\_PRICE attributes for drugs whose **price is less than 50.00**. Enter the values in the table below. Add rows to the table if needed. (can use SQL or manually determine the results)

|  |  |
| --- | --- |
| DRUG\_CODE | DRUG\_PRICE |
| AF15 | 25.00 |
| AF25 | 35.00 |
| DRZ | 18.99 |

1. (2 pts) Create the table that would result from applying the SELECT and PROJECT relational operators to the DRUG table to return only the DRUG\_CODE and DRUG\_PRICE attributes for drugs whose **price is greater than 30.00 and less than 80.00**. Enter the values in the table below. (can use SQL or manually determine the results)

|  |  |
| --- | --- |
| DRUG\_CODE | DRUG\_PRICE |
| AF25 | 35.00 |
| KO15 | 65.75 |
| TRVP | 79.45 |

1. (2 pts) Create the table that would result from applying a **DIFFERENCE** relational operator of your result from problem 10 to your result from problem 11. Enter the values in the table below. (can use SQL or manually determine the results)

|  |  |
| --- | --- |
| DRUG\_CODE | DRUG\_PRICE |
| AF15 | 25.00 |
| DRZ | 18.99 |