Kevin Hance
DBMS (CPSC 321-01)
Due 10/1/19
HW4

READING ASSIGNMENT

1. The basic types in the textbook consist of the following:
   char, varchar, int, smallint, numeric, real/double, and float
   MariaDB has full support for all of these types, as well as many more. These include but are not limited to the following:
   binary/charbyte, varbinary, tinyblob, blob, mediumblob, longblob, tinytext, text, mediumtext, longtext, JSON type, enum, date, time, datetime, timestamp, year, boolean, mediumint, tinyint, big int, decimal, fixed, and bit.
   In addition to these, MariaDB supports functionality to save row procedure operations and BLOB operations. Lastly, some geometrical types are supported by MariaDB. This is done by describing points, lines, and polygons as Strings.

2. The purpose of NOT_NULL is to ensure that value never has a NULL value. Primary keys are always NOT_NULL by default. NOT_NULL is oftentimes used because, in the context of the relational table, a NULL value would cause a problem. An example of this being used would be in a table keeping track of online orders. Most online services will allow the user to input an Apartment or Suite Number if they live in an apartment complex. This value could remain NULL, as many people do not live in an apartment. However, the street address would be made NOT_NULL because anyone capable of receiving mail would need to have a valid street address at a minimum.

user_id	***order_id***	*product_id*	*street_address*	*apartment_number*
J39A42K19L	000001856291	00F3J9A	123 45th St.	450B
H19S9AG9K	000002391925	00FKL93	135 79th Ave.	NULL

   In this table, *user_id* and *order_id* are NOT_NULL by default because they make up a compound primary key. *product_id* is NOT NULL because a shipping order must contain a product to deliver, and the *street_address* is NOT_NULL because an order requires an address to deliver to. The *apartment_number*, however, is not constrained to be NOT_NULL because not all addresses must contain an apartment number.

3. I tested ‘ALTER TABLE’ by creating a new table named *test* with one attribute as the primary key. I then added a second attribute *major* of type VARCHAR(10) to the table, and the Query returned as OK, which implies the command worked as expected. I then added an extra value with a non-null value to test that the *major* attribute worked. My screencaps are shown below.

```
mysql> CREATE TABLE test(
    -> name VARCHAR(10),
    -> PRIMARY KEY(name)
    -> ) ENGINE=InnoDB;
Query OK, 0 rows affected (0.04 sec)

mysql> show tables;
+---------------------+
| Tables_in_khance_DB |
+---------------------+
| test                |
+---------------------+
1 row in set (0.00 sec)

mysql> INSERT INTO test VALUES ('Kevin'), ('Bassel'), ('Zachary');
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> show tables
    -> ;
+---------------------+
| Tables_in_khance_DB |
+---------------------+
| test                |
+---------------------+
1 row in set (0.00 sec)

mysql> alter table test add major VARCHAR(10);
Query OK, 0 rows affected (0.00 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> INSERT INTO test VALUES  ('KevinH', 'CS');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> SELECT * FROM test;
+---------+-------+
| name    | major |
+---------+-------+
| Bassel  | NULL  |
| Kevin   | NULL  |
| KevinH  | CS    |
| Zachary | NULL  |
+---------+-------+
4 rows in set (0.00 sec)
```

4. The result of the CREATE OR REPLACE TABLE command was exactly what I expected. It replaced the table I had and removed all the values I had previously added. The rows of the table were removed, and I inserted new values to the new table.

```
mysql> SELECT * FROM test;
+---------+-------+
| name    | major |
+---------+-------+
| Bassel  | NULL  |
| Kevin   | NULL  |
| KevinH  | CS    |
| Zachary | NULL  |
+---------+-------+
4 rows in set (0.00 sec)

mysql>
mysql> show tables
    -> ;
+---------------------+
| Tables_in_khance_DB |
+---------------------+
| test                |
+---------------------+
1 row in set (0.00 sec)

mysql> SELECT * FROM test
    -> ;
+---------+-------+
| name    | major |
+---------+-------+
| Bassel  | NULL  |
| Kevin   | NULL  |
| KevinH  | CS    |
| Zachary | NULL  |
+---------+-------+
4 rows in set (0.00 sec)

mysql> CREATE OR REPLACE TABLE test(
    -> name VARCHAR(100),
    -> major VARCHAR(50),
    -> total_credits INT,
    -> PRIMARY KEY(name)
    -> ) ENGINE=InnoDB;
Query OK, 0 rows affected (0.07 sec)

mysql> select * from test
    -> ;
Empty set (0.00 sec)

mysql> INSERT INTO test VALUES ('Kevin Hance', 'Computer Science', 109), ('Bassel Mufarreh', 'Human Phys', 112), ('Zachary McKee', 'Computer Science', 121);
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from test;
+----------------+------------------+---------------+
| name           | major            | total_credits |
+----------------+------------------+---------------+
| Bassel Mufarreh | Human Phys       |           112 |
| Kevin Hance    | Computer Science |           109 |
| Zachary McKee  | Computer Science |           121 |
+----------------+------------------+---------------+
3 rows in set (0.00 sec)
```

TECHNICAL WORK

```
mysql> show tables
    -> ;
+--------------------+
| Tables_in_khance_DB |
+--------------------+
| album              |
| genre              |
| music_artist       |
| music_group        |
| record_label       |
| record_label_type  |
+--------------------+
6 rows in set (0.00 sec)

mysql> select * from album;
+---------------+--------------+------------------+---------------------------------+------------------------+
| title         | music_group  | year_of_recording | songs                           | record_label           |
+---------------+--------------+------------------+---------------------------------+------------------------+
| Great Album   | MusicMakers  |             2010 | song1, song2, song3, song4, song5 | Famous Label           |
| Rap Album     | The Rappers  |             1999 | beat1, beat2, beat3, beat4, beat5 | Real Ones Entertainment |
| Unknown Album | Beat Creators |             2009 | songA, songB, songC, songD, songE | Underground Label      |
+---------------+--------------+------------------+---------------------------------+------------------------+
3 rows in set (0.00 sec)

mysql> select * from music artist;
ERROR 1146 (42S02): Table 'khance_DB.music' doesn't exist
mysql> select * from music_artist;
+------------+-----------+------------+------------------+--------------+
| first_name | last_name | birth_year | range_of_activity | music_group  |
+------------+-----------+------------+------------------+--------------+
| Jack       | Peterson  |       1970 | 1999-present      | The Rappers  |
| Jim        | Bean      |       1974 | 2009-present      | Beat Creators |
| John       | Doe       |       1984 | 2005-present      | MusicMakers  |
+------------+-----------+------------+------------------+--------------+
3 rows in set (0.00 sec)

mysql> select * from music_group;
+--------------+-----------------+-----------+
| group_name   | year_of_founding | genre     |
+--------------+-----------------+-----------+
| Beat Creators |            2008 | indie-rock |
| MusicMakers  |            2005 | rock      |
| The Rappers  |            1998 | hip-hop   |
+--------------+-----------------+-----------+
3 rows in set (0.00 sec)

mysql> select * from genre;
+-----------+
| genre     |
+-----------+
| hip-hop   |
| indie-rock |
| rock      |
+-----------+
3 rows in set (0.00 sec)
```

```
mysql> select * from record_label;
+-------------------------+------------------+---------------+
| label_name              | year_of_founding | label_type_id |
+-------------------------+------------------+---------------+
| Famous Label            |             2000 |             0 |
| Real Ones Entertainment |             1997 |             2 |
| Underground Label       |             2013 |             1 |
+-------------------------+------------------+---------------+
3 rows in set (0.00 sec)

mysql> select * from record_label_type;
+---------------+------------+
| label_type_id | label_type |
+---------------+------------+
|             0 | major      |
|             1 | indie      |
|             2 | hip-hop    |
+---------------+------------+
3 rows in set (0.00 sec)

mysql>
```