Leandro Lopez
CSC 6302 Database Principles and Design
Merrimack College

Quiz 1

1) Physical Level: The lowest level of abstraction, the physical level, describes “how” the data are stores. Details at the physical level include things like the actual storage devices the data are stored in, how the data are stored in files, as well as the index of data. Database administrators and system administrators may interact with this level.

Logical Level: The middle level of abstraction, the logical level, describes the “what.” At this level, database administrators decide what the data looks like. This includes the data types as well as their relationship to each other. The complications of the physical level are hidden at this level, allowing users to work with the data without needing to understand how the database system is implemented. Application programmers and database administrators usually work at this level.

View Level: The highest level, the view level, provides parts of the database most relevant to the users need. For instance, when accessing an ATM, users only access their own data and not that of the banks other customers. While this level makes it easy to interact with the database, with the complexities of the logical level and the physical level hidden away, it also prevents users from entering unauthorized parts of the database.

2)
a.) The Material relation has **7 superkeys.**

b.)
1. MaterialName, Color, IsMachineWashable
2. MaterialName, Color
3. MaterialName, IsMachineWashable
4. Color, IsMachineWashable
5. MaterialName
6. Color
7. IsMachineWashable

c.)
1. MaterialName
2. MaterialName, Color

d.)
Primary key: MaterialName

e.)
```
CREATE TABLE IF NOT EXISTS Material (
   MaterialName VARCHAR(100) PRIMARY KEY,
   Color VARCHAR(50),
   IsMachineWashable TINYINT
);
```

3)

```
CREATE TABLE IF NOT EXISTS SewingPattern (
   PatternName VARCHAR(100) PRIMARY KEY,
   PublisherName VARCHAR(200),
   SkillLevel INT,
   MaterialName VARCHAR(100),
   Yardage DOUBLE,
   FOREIGN KEY(MaterialName) REFERENCES Material(MaterialName)

);
```

4)

a.) SELECT GameName FROM Game;

b.)

```
SELECT
        p.UserName,
        g.GameName
FROM
        Player p
JOIN
        Game g on p.FavoriteGame = g.Id;
```

5)

a.)

```
SELECT
        g.GameName,
        g.DeveloperName
FROM Game g
ORDER BY g.DeveloperName, g.GameName;
```

b.)

```
SELECT COUNT(DISTINCT Playerid) AS NumberOfPlayers
FROM Score
WHERE Score = 185000;
```

c.)

```
SELECT
        GameId,
        MAX(Score) AS HighestScore
FROM Score
GROUP BY GameId;
```